TUGAS AKHIR – TI 184833

ANALISA PERBANDINGAN METODE PREDIKSI PELANGGAN *CHURN* DENGAN PENERAPAN DATA MINING

ACH. NAFILA ROZIE
NRP 02411540000113

Dosen Pembimbing
Prof. Ir. Budi Santosa, MS., Ph.D
NIP. 196905121994021001

DEPARTEMEN TEKNIK INDUSTRI
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya
2019

(Halaman ini sengaja dikosongkan)

FINAL PROJECT – TI 184833

COMPARISON ANALYSIS OF CUSTOMER CHURN PREDICTION METHOD USING DATA MINING

ACH. NAFILA ROZIE
NRP 02411540000113

Supervisor
Prof. Ir. Budi Santosa, MS., Ph.D
NIP. 196905121994021001

INDUSTRIAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Institut Teknologi Sepuluh Nopember
Surabaya
2019

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

ANALISA PERBANDINGAN METODE PREDIKSI PELANGGAN *CHURN* DENGAN PENERAPAN DATA MINING

TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Teknik
pada Program Studi S-1 Departemen Teknik Industri
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya

Penulis:
ACH. NAFILA ROZIE
NRP 02411540000113

Disetujui oleh
Dosen Pembimbing Tugas Akhir:

Pembimbing Utama

Prof. Ir. Budi Santosa, MS., Ph.D
NIP. 196905121994021001

SURABAYA, JANUARI 2019

v

(Halaman ini sengaja dikosongkan)

ANALISA PERBANDINGAN METODE PREDIKSI PELANGGAN *CHURN* DENGAN PENERAPAN DATA MINING

Nama : Ach. Nafila Rozie
NRP : 02411540000113
Departemen : Teknik Industri – ITS
Pembimbing : Prof. Ir. Budi Santosa, MS., Ph.D

ABSTRAK

Pelanggan merupakan salah satu unsur penting dalam memastikan suatu preses bisnis perusahaan agar dapat terus bertahan serta bersaing dengan kompetitor. Berdasarkan kondisi dilapangan, pelanggan dapat sepenuhnya loyal menggunakan layanan jasa/produk suatu perusahaan atau berpindah menggunakan layanan jasa/produk dari perusahaan lain (*churn*). Hasil penelitian juga menunjukkan bahwa dengan menurunkan tingkat pelanggan berhenti menggunakan layanan jasa/produk dari suatu perusahaan dapat meningkat pendapat perusahaan hingga 95%. PT X sebagai perusahaan yang bergerak dibidang telekomunikasi memandang bahwa mempertahankan pelanggan jauh lebih penting mengingat biaya yang dibutuhkan lebih murah jika dibandingkan dengan mengakuisisi pelanggan baru. Maka dari itu diperlukan suatu sistem prediksi pelanggan yang mampu mengidentifikasi apakah suatu pelanggan berhenti menggunakan layanan dari PT X atau tidak. Hal tersebut dapat dicapai salah satu caranya dengan pendekatan *data mining.* Teknik data mining yang dipilih untuk membangun model prediksi pada penelitian ini adalah teknik klasifikasi dengan Regresi Logistik, *Naïve Bayes* dan *Support Vector Machine*. Dari hasil komparasi model, diperoleh bahwa model terbaik menggunakan pendekatan *Support Vector Machine* dengan kernel RBF. Akurasi serta *recall* dari model tersebut dalam memprediksi pelanggan *churn* masing-masing mencapai 89.63% dan 89.79%. Uji coba sistem yang dilakukan menggunakan data pelanggan dari PT X di area kerja Bogor.

Kata kunci: *Data Mining, Churn,* Regresi Logistik*, Naïve Bayes Classifier, Support Vector Machine*

(Halaman ini sengaja dikosongkan)

COMPARISON ANALYSIS OF CUSTOMER CHURN PREDICTION METHOD USING DATA MINING

Name : Ach. Nafila Rozie
NRP : 02411540000113
Department : Industrial Engineering – ITS
Supervisor : Prof. Ir. Budi Santosa, MS., Ph.D

ABSTRACT

The customer is one of the important elements in ensuring a company can survive and compete with competitors. Based on the field conditions, customers can be fully loyal to use the services / products of a company or move to use services / products from other companies (churn). The results of the study also show that by reducing the level of customers stopping using services / products from a company can increase the company's profit up to 95%. PT X as a telecommunications company considers that retaining customers is far more important compared to acquiring new customers. Therefore, they need a customer prediction system that is able to identify whether a customer has stopped using services from PT X or not. This can be achieved by using a data mining approach. The data mining techniques chosen to build the prediction model in this study are classification techniques with Logistic Regression, Naïve Bayes and Support Vector Machine. From the results of the model comparison and parameters tuning, it was found that the best model used the Support Vector Machine with RBF kernel approach. Accuracy and recall values of the model reaching 89.63% and 89.79% respectively. System testing is carried out using customer data from PT X in the Bogor work area.

Keywords: Data Mining, Churn, Logistic Regression, Naïve Bayes Classifier, Support Vector Machine

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Bismillahirrahmanirrahim.

Puji dan syukur penulis panjatkan ke hadirat Allah SWT karena atas limpahan rahmat dan karunia-Nya penulis dapat menyelesaikan Laporan Tugas Akhir ini dengan lancar dan tepat pada waktunya. Shalawat serta salam juga senantiasa penulis ucapkan kepada Nabi Muhammad SAW. Laporan Tugas Akhir ini disusun untuk memenuhi persayaratan dalam menyelesaikan studi Strata-1 di Departemen Teknik Industri, Institut Teknologi Sepuluh Nopember Surabaya. Selama proses pengerjaan laporan, banyak hambatan yang penulis alami. Bantuan, saran dan dukungan motivasi dari berbagai pihak sangat membantu dalam penyelesaian Laporan Tugas Akhir ini. Pada kesempatan ini penulis ingin menyampaikan ucapan terimakasih kepada seluruh pihak yang telah membantu dalam penyelesaian laporan ini, yaitu:

1. Allah SWT yang senantiasa melindungi dan memberikan petunjuk dan kemudahan kepada penulis dalam menyelesaikan laporan Tugas Akhir ini.
2. Bapak Prof. Ir. Budi Santosa, MS., Ph.D selaku dosen pembimbing yang selalu memberikan bimbingan, arahan, nasihat, dan motivasi dalam penyelesaian Tugas Akhir ini.
3. Bapak Firmanda Robi yang telah memberikan kesempatan, bantuan, serta bimbingan dalam proses pengambilan data di perusahaan.
4. Bapak Erwin Widodo, ST., M.Eng., Dr.Eng., dan Bapak Dr. Eng. Ir. Ahmad Rusdiansyah, MEng selaku dosen penguji Tugas Akhir yang telah banyak memberi saran dan masukan untuk perbaikan Tugas Akhir ini.
5. Seluruh Bapak dan Ibu dosen Departemen Teknik Industri ITS yang telah mendidik dan mengajarkan berbagai ilmu selama masa perkuliahan sebagai bekal di kemudian hari.
6. Ibu tercinta serta keluarga besar yang selalu memberikan dukungan, semangat, dan motivasi yang sangat luar biasa kepada penulis.
7. Teman-teman TI angkatan 2015 (ICARUS) dan teman-teman organisasi yang menjadi teman seperjuangan selama masa perkualiahan yang selalu memberikan keceriaan dan motivasi dalam penyelesaian Tugas Akhir.
8. Semua pihak yang terlibat yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa penulisan laporan Tugas Akhir ini tidak lepas dari kesalahan dan kekurangan. Oleh karena itu, penulis mohon kritik dan saran pembaca yang dapat membangun dan memperbaiki penulisan selanjutnya.

Surabaya, 15 Januari 2019

Penulis

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

BAB 1
PENDAHULUAN

Pada bab ini akan dijelaskan mengenai pendahuluan penelitian ini meliputi latar belakang, rumusan masalah, tujuan, manfaat, serta ruang lingkup penelitian.

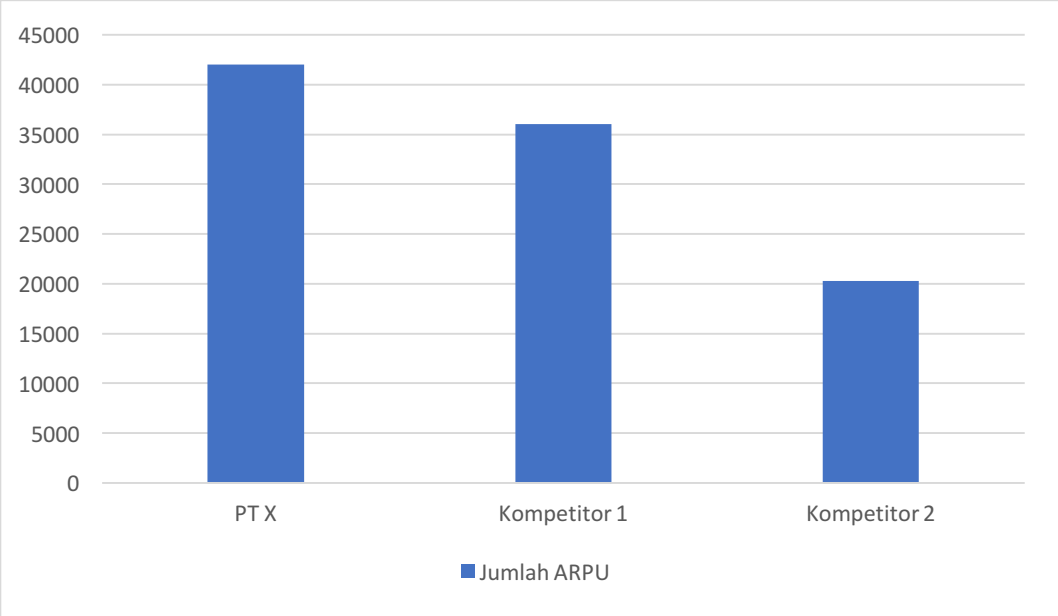
1.1 Latar belakang

Di era revolusi industri 4.0 saat ini, persaingan usaha yang semakin kompetitif mendorong industri mencari berbagai cara untuk mampu tetap bertahan, tidak terkecuali bagi industri yang bergerak dibidang telekomunikasi. Hal tersebut mengindikasikan bahwa adanya probabilitas pelanggan berpindah layanan terhadap suatu produk semakin besar. Pelanggan *churn* merupakan berpindahnya pelanggan dalam menggunakan jasa suatu *provider* (perusahaan penyedia jasa telekomunikasi) ke *provider* lainnya (Wei dan Chiu, 2002). Sedangkan faktor penyebab pelanggan merubah produk/ jasa yang digunakan beberapa diantaranya disebabkan oleh faktor kualitas, biaya, hingga teknologi yang digunakan oleh kompetitor (Ahn dkk, 2006).

Maka dari itu beberapa perusahaan dibidang penyedia layanan jasa, khususnya telekomunikasi mengalihkan fokus strategi mereka dari akuisisi pelanggan ke retensi pelanggan (Venkatesan and Kumar, 2002). Hal tersebut dilatarbelakangi oleh perusahaan dapat meningkatkan rata-rata pendapatan bersih hingga 95% hanya dengan menurunkan tingkat *churn* sebesar 5%. Terlebih pada perusahaan dengan tingkat *churn* tahunan yang tinggi (20 – 40%), sudah seharusnya perusahaan tersebut memiliki program manajemen pelanggan *churn* yang baik untuk mengoptimalkan pendapatan mereka (Kim dkk, 2004; Eshgi dkk, 2007).

Alasan lain yang mendorong perusahaan penyedia layanan jasa telekomunikasi perlu mengambil tindakan guna menekan tingkat *churn* yang ada pada pelanggannya adalah terdapatnya *potential loss* sehingga dapat menggerus laba perusahaan. *Potential loss* pada kasus ini dapat ditentukan dengan mengalikan jumlah pengguna, tingkat *churn,* dengan tingkat ARPU (*Average Revenue per User*) pada suatu perusahaan. ARPU merupakan keutungan perbulan yang didapat perusahaan dibagi dengan jumlah *user* yang menggunakan layanan perusahaan

tersebut (Rice, 2014). Mengacu pada laporan tahunan perusahaan, PT X telah memiliki 173.92 juta pengguna di awal tahun 2017 dengan *market share* di kawasan Jabodetabek Jabar (Jakarta, Bogor, Depok, Tangerang, Bekasi, dan Jawa Barat) sendiri telah mencapai 33%. Disisi lain, berdasarkan riset yang dilakukan Bank of America Merrl Lynch (BoAML) pada tahun 2015 mengatakan bahwa rata-rata *churn rate* perusahaan telekomunikasi yang ada di Indonesia berada di kisaran angka 10 persen/bulan (Zaenuddin, 2017). Berdasarkan Gamba 1.1, diperoleh informasi bahwa PT X merupakan salah satu perusahaan yang memiliki dampak paling signifikan akibat *churn* karena memiliki tingkat ARPU tertinggi dibandingkan kompetitor-kompetitornya. Dengan tingkat ARPU, *churn,* jumlah pengguna di kawasan tersebut, *potential loss* yang diterima oleh PT X mencapai 241.48 miliar rupiah per bulan.



Gambar 1.1 Jumlah ARPU (Average Revenue per User) pada 3 Industri Telekomunikasi di Indonesia (dalam rupiah) Tahun 2017. (Sumber: Tirto.id)

Dari kedua isu diatas, PT X mulai menyadari pentingnya mengatasi masalah *churn* pada pelanggannya. Beberapa perusahaan penyedia jasa telekomunikasi di Indonesia saat ini mulai menerapkan teknik-teknik *data mining* untuk mengakomodasi isu-isu yang berhubungan dengan *churn* yang terdapat pada pelanggannya, salah satunya adalah dengan membangun model *churn prediction*. Sebagai salah satu perusahaan dengan *market share* terbesar di Indonesia, perlu

untuk dilakukan usaha-usaha untuk mempertahankan serta meningkatkan ARPU serta *market share* yang telah diraih.

PT X telah memiliki basis data untuk menyimpan data-data yang dibutuhkan dalam menjalankan bisnisnya. Data-data berjumlah besar yang tersimpan secara elektronik pada basis data PT X dan dapat dimanfaatkan untuk menemukan pola-pola perilaku serta karakteristik yang ada pada pelanggannya. Pengolahan yang tepat pada data-data tersebut dapat menghasilkan pengetahuan-pengetahuan yang bermanfaat untuk memahami pola perilaku *churn* serta memprediksi pelanggan mana yang berpotensi akan meninggalkan perusahaan. Proses analisis data untuk menemukan informasi dan pengetahuan pada data yang sangat besar disebut *data mining*.

Data mining sebagai salah satu alat analisis pada CRM (*Customer Relationship Management*), memiliki banyak metode alternatif yang dapat diterapkan untuk memperoleh informasi dari data yang ada. Dalam penerapannya, metode yang terdapat dalam *data mining* diperlukan adaptasi setiap periode waktu tertentu (Rodpysh et.al, 2002). Hal ini bertujuan agar metode yang digunakan secara konstan mampu menghasilkan prediksi dengan akurasi yang baik. Selain itu, metode yang paling optimal dalam memprediksi pelanggan *churn* pada periode waktu saat ini belum menjadi jaminan menjadi yang paling optimal untuk periode waktu selanjutnya sehingga dibutuhkan adaptasi secara kontinu (Santosa, 2018).

Oleh karena itu, *data mining* dapat digunakan sebagai salah satu pendekatan untuk menjawab kebutuhan analisis dan prediksi dalam memecahkan masalah *churn* yang muncul pada industri telekomunikasi, khususnya pada PT X. Berdasarkan uraian tersebut serta mengingat bahwa prediksi *churn* merupakan aktivitas yang dilakukan berulang-ulang, dapat ditarik kesimpulan bahwa PT X membutuhkan suatu sistem prediksi *churn* untuk menyediakan pengetahuan dan informasi yang mendukung pembuatan program-program retensi serta mengurangi jumlah pelanggan yang *churn*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah disebutkan sebelumnya, rumusan masalah yang akan diangkat dalam penelitian ini adalah bagaimana membuat

sebuah model *churn prediction* untuk pelanggan dalam melakukan prediksi apakah suatu pelanggan dikategorikan sebagai *churn* atau *non-churn* sehingga mendapat perlakuan khusus.

1.3 Tujuan

Adapun tujuan dari penelitian tugas akhir ini adalah sebagai berikut.

1. Membangun model *churn prediction* dengan pendekatan *data mining.*
2. Mengetahui preferensi teknik yang lebih baik dalam melakukan prediksi *churn*.
3. Menentukan variabel yang paling signifikan terhadap penentuan seorang pelanggan dikategorikan *churn* atau *non-churn.*

1.4 Manfaat Penelitian

Adapun manfaat yang dapat diperoleh dengan diadakannya penelitian ini adalah sebagai berikut:

1. Mendapatkan model *churn prediction* dengan menggunakan teknik *data mining.*
2. Mengetahui lebih banyak preferensi teknik dalam melakukan prediksi pelanggan *churn.*
3. Mengetahui variabel yang paling signifikan dalam menentukan seorang pelanggan dikategorikan *churn* atau *non-churn*.

1.5 Ruang Lingkup Penelitian

Ruang lingkup dalam penelitian dijelaskan melalui dua bagian yaitu batasan dan asumsi. Berikut adalah batasan dan asumsi yang digunakan.

1.5.1 Batasan

Berikut adalah batasan yang digunakan dalam menyelesaikan tugas akhir ini.

1. Penelitian dilakukan pada PT X yang bergerak dibidang penyedia jasa telekomunikasi
2. Kasus yang diangkat terbatas untuk klasifikasi dua kelas dengan output adalah pelanggan *churn* dan *non-churn*

3. Data yang diambil merupakan data pelanggan PT X di area kerja Bogor
4. Kerangka waktu data pelanggan yang digunakan adalah sejak bulan Januari 2017 hingga bulan Desember 2018.

1.5.2 Asumsi

Berikut adalah asumsi yang digunakan dalam menyelesaikan tugas akhir ini.

1. Tidak ada perubahan *churn rate* sejak tahun 2015 -2017
2. Pelanggan yang dikategorikan sebagai *churn* merupakan pelanggan yang memiliki status sebagai *inactive* atau *grace* dengan durasi lebih dari 30 hari.

1.6 Sistematika Penulisan

Sistematika penulisan pada penelitian ini terdiri dari enam bab. Berikut merupakan garis besar penelitian yang dilakukan.

BAB 1 PENDAHULUAN

Pada bagian ini dijelaskan mengenai latar belakang dilakukannya penelitian ini, rumusan masalah yang akan diselesaikan, tujuan, manfaat, ruang lingkup penelitian yang terdiri dari asumsi dan batasan, dan adanya penjelasan mengenai sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Pada bagian ini dijelaskan mengenai berbagai landasan teori yang digunakan untuk menyelesaikan permasalahan. Pembahasan kajian teori yang dilakukan adalah terkait dengan *customer relationship management*, *churn, data mining, k-fold cross validation*, regresi logistik, *naïve bayes, support vector machine*, serta kinerja metode. Selain itu, pada bab ini juga dijelaskan mengenai letak perbedaan antara penelitian ini dengan penelitian yang telah dilakukan sebelumnya yang terkait dengan prediksi pelanggan *churn*. Selain itu, pada bab ini juga akan dijelaskan mengenai perbedaan yang membedakan penelitian ini dengan penelitian-penelitian sebelumnya yang berhubungan dengan prediksi *churn*.

BAB III METODOLOGI PENELITIAN

Pada bagian ini dijelaskan mengenai tahapan proses pengerjaan penelitian yang terdiri dari *flow chart* proses dan penjelasannya. Selain itu, pada bagian *data*

processing akan dijelaskan untuk setiap proses pada pembuatan model dengan pendekatan regresi logistik, *support vector machine*, dan *naïve bayes*.

BAB IV PENGUMPULAN DAN PENGOLAHAN DATA

Pada bagian ini dijelaskan mengenai data yang dikumpulkan untuk menyelesaikan permasalahan yang sedang diteliti. Selain pengumpulan data, pengolahan juga dilakukan untuk memperoleh data yang siap digunakan untuk membangun model.

BAB V PEMBANGUNAN DAN ANALISA MODEL

Pada bagian ini dijelaskan mengenai pembangunan serta analisis terhadap masing-masing model. Pembangunan model berkaitan dengan data yang telah dilakukan sebelumnya sedangkan analisis yang dilakukan terkait dengan hasil model yang telah dibangun dengan pendekatan tiga metode yang berbeda untuk selanjutnya menjadi sebuah solusi yang ditawarkan kepada PT X.

BAB VI KESIMPULAN DAN SARAN

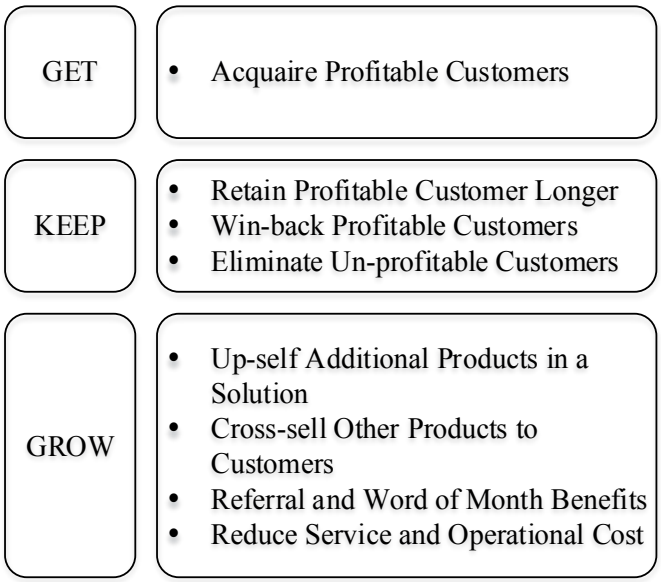
Pada bagian ini dijelaskan kesimpulan dari penelitian dan saran yang diberikan terkait penelitian yang telah diselesaikan.

BAB 2
TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai literatur dan kajian pustaka terkait dasar ilmu untuk pelaksanaan penelitian ini.

2.1 Customer Relationship Management

Menurut Bligh & Douglas (2004), *Customer Relationship Management* atau CRM merupakan kombinasi dari kebijakan, proses, dan strategi yang diimplementasikan oleh sebuah perusahaan yang berfokus pada interaksi pelanggan dengan menyediakan sebuah mekanisme *tracking* informasi mengenai pelanggan.



Gambar 2. 1 Diagram Get-Keep-Grow Pelanggan (Sumber: Peppers & Rogers, 2004)

Menurut Peppers & Rogers (2004), seperti yang terdapat dalam Gambar 2.1, *goal* dari setiap perusahaan adalah sesuatu yang sederhana, yaitu *get-keep-grow* pelanggan. CRM mampu mengakomodasi perusahaan dalam menyediakan *excellent service* terhadap pelanggan, dengan cara *develop* hubungan dengan tap-tiap *valued customers.* Untuk membangun *relationship ini,* CRM menggunakan informasi dari setiap *individual account* secara efektif.

Menurut Kotler (2003), perusahaan-perusahaan tersebut meningkatkan *value* pelanggannya dengan beberapa strategi berikut ini:

1. Menurunkan tingkat pelanggan yang berpindah
2. Meningkatkan kualitas *customer relationship*
3. Meningkatkan *growth potential* dari masing-masing pelanggan melalui *"share of wallet"*
4. Mengusahakan pelanggan untuk segmen *low profit* menjadi lebih menguntungkan, atau menghilangkan mereka.
5. Berfokus untuk memberikan *extra effort* kepada *high value customers*.

2.2 *Churn*

Churn pada pelanggan didefinisikan sebagai kecenderungan pelanggan dalam menghentikan kontak dengan perusahaan (Yang dan Chiu, 2006). Selain itu, menurut Yang dan Chiu, pelanggan *churn* dibagi menjadi tiga tipe yaitu: *Involuntary churn* (terjadi ketika pelanggan gagal membayar tagihannya, sehingga perusahaan memutuskan untuk mencabut layanannya), *Inevitable churn* (terjadi ketika pelanggan mati atau bermigrasi sehingga menghilangkan pelanggan sepenuhnya dari pasar), dan *voluntary churn* (terjadi ketika pelanggan secara sadar memilih untuk beralih menggunakan layanan operator lain karena lebih bernilai). Dari ketiga jenis pelanggan *churn* tersebut, *involuntary churn* dan *inevitable churn* merupakan kerjadi yang tidak dapat perusahaan cegah, namun untuk *voluntary churn*, perusahaan dapat berupaya ikut andil dalam menekan pelanggan melakukan *churn* ke *provider* atau penyedia jasa layanan telekomunikasi lain.

Berikut merupakan kategori variabel yang secara signifikan mempengaruhi pelanggan untuk *churn* (Ahn dkk, 2006), diantaranya:

1. *Membership card* (kartu atau fasilitas anggota)

   Membership card atau kartu atau fasilitas anggota merupakan segmentasi yang perusahaan penyedia layanan jasa telekomunikasi tawarkan dengan keuntungan tertentu. Contohnya mulai dari: menawarkan banyak kelebihan namun terbatas untuk *heavy-users*, menawarkan sedikit kelebihan yang namun terbuka untuk semua pemilik *membership card*, dan tidak menawarkan kelebihan bagi yang tidak memiliki *membership card*.

2. *Call drop rate* (rasio panggilan putus)

   Call drop rate atau rasio panggilan putus merupakan proporsi panggilan putus yang pelanggan alami terhadap total panggilan yang dicoba.

3. *Call failure rate* (rasio panggilan gagal)

   Call failure rate atau rasio panggilan gagal merupakan proporsi panggilan gagal yang pelanggan alami terhadap total panggilan yang dicoba.

4. *Number of complaints* (jumlah komplain)

   Number of complaints atau jumlah komplain merupakan frekuensi pelanggan pengguna jasa telekomunikasi melakukan complain terhadap *customer service* terkait masalah dengan pembayaran, fasilitas, hingga layanan.

5. *Loyalty points* (loyalitas pelanggan)

   Loyalty points atau loyalitas pelanggan merupakan jumlah kredit yang pelanggan keluarkan, dimana kredit tersebut mampu ditukarkan menjadi berbagai macam produk dan/atau jasa.

6. *Billed amounts* (jumlah tagihan)

   Billed amounts atau jumlah tagihan merupakan total tagihan bulanan yang pelanggan harus bayar.

7. *Unpaid balances* (tagihan yang tidak terbayar)

   Unpaid balances atau tagihan yang tidak terbayar merupakan total tagihan yang tidak dibayar oleh pelanggan.

8. *Number of unpaid monthly bills* (Frekuensi tagihan bulanan yang tidak terbayar tepat waktu)

   Number of unpaid monthly bills atau frekuensi tagihan bulanan yang tidak terbayar tepat waktu merupakan frekuensi dimana pelanggan tidak membayar tagihan bulanannya sebelum atau tepat pada tenggat pembayaran yang telah ditentukan.

9. *Gender* (jenis kelamin)

   Gender atau jenis kelamin merupakan keterangan jenis kelamin bagi pengguna jasa layanan telekomunikasi.

10. *Call plans* (rencana panggilan)

    Call plans atau rencana panggilan di desain untuk menyesuaikan dengan *behavior* kebutuhan pelanggan dalam menggunakan jasa telekomunikasinya. Contohnya mulai dari penerapan sistem pra-bayar, bayar, dan pasca-bayar untuk menyesuaikan dengan kebutuhan pelanggan.

11. *Handset internet capability* (kapabilitas perangkat internet)

    Handset internet capability atau kapabilitas perangkat internet merupakan jenis perangkat elektronik telekomunikasi yang digunakan oleh pelanggan.

12. *Customer Status* (status pelanggan)

    Customer Status atau status pelanggan merupakan jumlah pelanggan yang tidak aktif, *suspended*, serta aktif.

Dari faktor-faktor yang yang mempengaruhi tingkat pelanggan *churn* di atas, selanjutnya dijadikan pertimbangan dalam menentukan data apa saja yang akan digunakan dalam melakukan pembuatan model yang dapat menlakukan klasifikasi terhadap suatu pelanggan apakah *churn* atau *non-churn.*

2.3 Data Mining

Data Mining atau Penggalian data adalah kegiatan mengekstrak informasi atau pengetahuan (*knowledge*) penting dari suatu set data berukuran besar dengan menggunakan teknik tertentu (Santosa, 2018). Informasi yang diperoleh diharapkan dapat menjadi pertimbangan untuk memperbaiki pengambilan keputusan di masa depan. Karena banyaknya data yang berbentuk elektronik dan kebutuhan pengubahan data tersebut menjadi informasi dan pengetahuan, penggalian data menjadi salah satu solusi yang dilirik di dunia informasi akhir-akhir ini. Secara garis besar, proses *knowledge discovery* adalah sebagai berikut (Han, 2011) :

1. *Data integration*, pada tahap ini data dari berbagai sumber dikumpulkan dan diintegrasikan menjadi satu.

2. *Data selection*, data yang berhubungan dengan tugas analisis diambil dari basis data.
3. *Data cleaning*, pada tahap ini dilakukan proses pembersihan data yang keliru, hilang, hingga yang tidak berhubungan terhadap variabel tujuan.
4. *Data transformation*, proses mengolah (mengubah atau menggabung) data menjadi bentuk yang sesuai untuk proses penggalian data dengan melakukan operasi agregasi dan perangkuman
5. Penggalian data, proses ini merupakan proses utama dari *knowledge discovery*, dimana metode diterapkan pada data untuk menghasilkan pola data
6. *Pattern Evaluation*, proses untuk mengidentifikasi pola yang menarik yang merepresentasikan pengetahuan berdasarkan *interestingness measure*.
7. *Knowledge presentation*, teknik visualisasi dan representasi pengetahuan digunakan untuk menyampaikan pengetahuan pada *user*.

Tugas penggalian data dapat dikategorikan menjadi dua bagian yaitu *descriptive* dan *predictive.* Penggalian data *descriptive* menjelaskan data set dengan ringkas dan menyampaikan sifat umum dari data yang telah di analisa. Sedangkan untuk,penggalian data *predective* membuat satu atau sekumpulan model yang kemudian diambil dugaan mengenai data set yang ada. Kemudian, penggalian data *predictive* akan memprediksi karakteristik *dataset* terbaru. Berikut merupakan beberapa tugas yang dapat dilakukan oleh *data mining* (Santosa, 2018):

1. *Clustering* (Pengelompokkan)

Clustering mengelompokkan obyek ke dalam beberapa kelompok berdasarkan kemiripan antar obyek, dimana dalam satu klaster harus berisi obyek yang memiliki karakteristik yang sama sedangkan antar klaster harus memiliki karakteristik yang berbeda. *Clustering* disini tidak membutuhkan *training data* yang telah diberi label (*ground truth label*).

2. *Classification* (Klasifikasi)

Klasifikasi melakukan pengelompokkan obyek berdasar kelompok data yang sudah ada. Berbeda dengan *clustering,* klasifikasi disini memerlukan *ground truth label.* Sebagai contoh, kita ingin mengelompokkan data gambar kanker ringan dan akut, maka kita menyiapkan misalnya 1000 gambar data pelatihan (*training*

data) dengan label kanker ringan dan 1000 gambar dengan label kanker akut. Prediksi pengelompokkan dilakukan dengan membangun model terlebih dahulu melalui proses pelatihan menggunakan data yang sudah ada. Kemudian, model yang telah terbentuk dari proses pelatihan dapat digunakan untuk mengelompokkan data baru.

3. Regresi/ Estimasi

Regresi pada dasarnya hampir serupa dengan klasifikasi, yakni membutuhkan data pelatihan yang telah diberi label. Namun, letak perbedaan antara regresi dan klasifikasi adalah, *output* dari klasifikasi adalah nilai diskret sedangkan *output* dari regresi adalah nilai kontinu. Regresi disini berperan dalam mencari relasi atau hubungan antara atribut prediktor (*independent*) dengan atribut respon (*dependent*), dimana atribut responnya juga berupa nilai kontinu. Contoh penerapan regresi adalah memprediksi nilai kurs rupiah terhadap nilai dollar.

4. *Association*

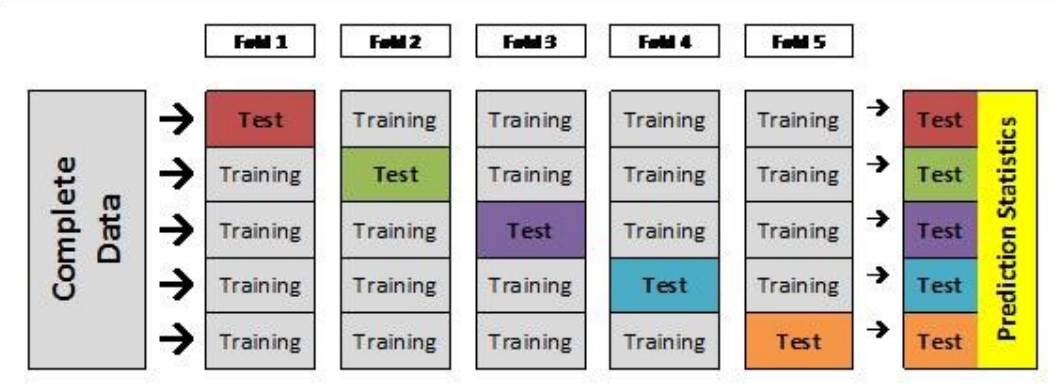
Melakukan asosiasi antar obyek dalam suatu set data, pada umumnya berupa data transaksional. Asosiasi dilakukan dengan menghitung berapa kali dalam sautu *dataset* suatu transaksi yang mengandung dua item atau lebih yang berhubungan. Tugas ini dapat pula disebut sebagai *Market Basket Analysis*.

Metode *data mining* yang digunakan pada penelitian ini untuk membantu pengolahan data *customer* dari PT X. Tugas masing-masing metode pada penelitian ini termasuk dalam *predictive data mining*. Hal ini dikarenakan penggalian data dapat digunakan untuk membangun model untuk melakukan prediksi data di masa depan. *Classification* adalah kategori tugas penggalian data yang dipilih pada penelitian ini.

2.4 K-Cross Fold Validation

K-Cross Fold Validation fold validation adalah suatu metode validasi yang umum digunakan dalam penelitian. Metode ini dilakukan dengan membagi data menjadi sejumlah *k* partisi, kemudian dilakukan percobaan sebanyak *k* kali untuk satu model dengan parameter yang sama (Santosa, 2018). Percobaan pertama yaitu menjadikan bagian pertama menjadai data *testing*, kemudian bagian sisanya menjadi data *training*. Percobaan kedua yaitu dengan menjadikan bagian kedua

menjadi data *testing*, kemudian bagian sisanya data *training*. Berlanjut hingga *k* kali percobaan.



Gambar 2. 2 Ilustrasi 5-*fold cross validation* (Sumber : Brinberg, 2015)

Secara umum, akan dibandingkan *n* model dalam *cross-validation* dan akan memilihh model yang paling bagus adalah model yang memberikan nilai rata-rata akurasi tertinggi dalam *cross-validation* tersebut. Untuk angka standar deviasi juga dapat dijadikan pertimbangan dalam mengetahui konsistensi sebuah model yang bersangkutan. Selain itu, metode perbandingan dapat dilakukan dengan menggunakan metode yang sama namun dengan menggunakan parameter yang berbeda. Misalkan pada metode *Support Vector Machine* dengan menggunakan kernel RBF. Model tersebut memliki dua parameter model, yaitu konstanta C untuk *slack* (berperan dalam mengatur *miss-classification*) dan nilai standar deviasi dari kernel RBF-nya.

2.5 Regresi Logistik

Regresi logistik atau model logistik atau model logit merupakan hubungan antara variabel independen dengan sebuah variabel dependen terkategori, serta mengestimasi sebuah kejadian dengan *fitting* terhadap kurva logistik. Regresi logistik memiliki 2 jenis model yaitu regresi logistik biner dan regresi logistik multinomial. Regresi logistik biner digunakan ketika variabel dependen bersifat dikotomi (memiliki 2 macam *output*). Sedangkan regresi logistik multinomial digunakan apabila variabel dependen memiliki lebih dari 2 macam *output* namun tetap terkategori (Park, 2013). Regresi logistik dapat digunakan pada bidang yang luas, contoh penerapannya dapat ditemukan mulai dari melakukan klasifikasi suatu pasien terdeteksi penyakit kanker atau tidak, melakukan klasifikasi gaji seorang pekerja masuk ke dalam tingkat berapa,

hingga melakukan klasifikasi seorang anak yang berada dalam kandungan ibu memiliki jenis kelamin laki-laki atau perempuan.

Berikut merupakan persamaan yang digunakan dalam menentukan fungsi regresi logistik biner (Gronros & Ida, 2018) adalah sebagai berikut:

$$p(X) = \frac{1}{1+ e^{-(\beta_0+\beta_1 X)}} \quad (2.1)$$

dan untuk kasus *multiple predictors* sebagai berikut:

$$p(X) = \frac{1}{1+ e^{-(\propto+\beta 1 X 1+\cdots+ \beta_k X_k)}} \quad (2.2)$$

Regresi logistik menghasilkan *sigmoid curve,* hal tersebut sekaligus menjadi jaminan bahwa *output* dari klasifikasi yang dilakukan terjadi pada interval yang dibutuhkan. Berikut merupakan perasaman regresi logistik biner setelah diberikan logaritma dikedua sisi:

$$\log\left(\frac{p(X)}{1- p(X)}\right) = \beta_0 + \beta_1 X \quad (2.3)$$

dan untuk kasus *multiple predictors* sebagai berikut:

$$\log\left(\frac{p(X)}{1- p(X)}\right) = \beta_0 + \beta_1 X + \cdots + \beta_k X_k \quad (2.4)$$

Parameter β_0 dan β_1 diestimasikan menggunakan *maximum likelihood* menggunakan fungsi dibawah.

$$Likelihood\ (\beta) = \prod_{i:y_i=1} p(x_i) + \prod_{i\prime:y_i\prime=0}(1 - p(x_i)) \quad (2.5)$$

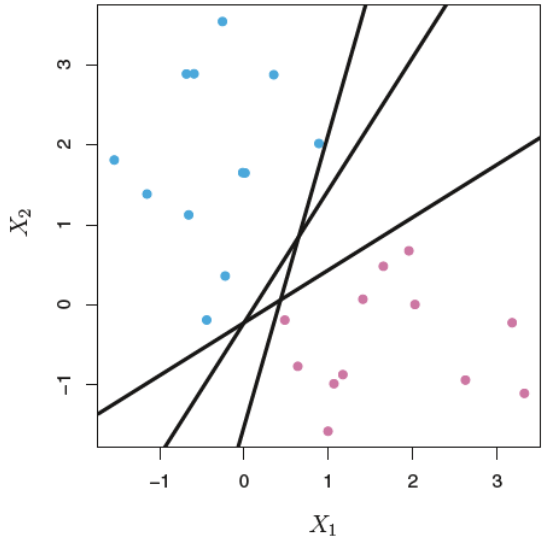
Sehingga apabila nilai beta telah diestimasikan, model dapat digunakan untuk prediksi terhadap data *testing*.

2.6 Support Vector Machine

Support Vector Machine atau SVM merupakan salah satu teknik yang relatif baru (1995) yang digunakan untuk melakukan prediksi yang berlaku baik untuk regresi, klasifikasi, hingga klasifikasi multi kelas (Santosa, 2018). SVM tergolong dalam *supervised learning,* dimana hal ini merupakan sebuah pendekatan yang membutuhkan data untuk dilatih serta variabel yang ditargetkan sehingga tujuan dari pendekatan metode ini adalah mengelompokkan suatu data ke dalam data yang sudah ada atau disebutkan sebelumnya(Chandra, 2017). SVM memiliki keluaran berupa bilangan bulat/ diskret.

Teknik SVM dapat diterapkan pada banyak bidang mulai dari: finansial, cuaca, pendidikan, kesehatan, hingga militer. Tujuan dari teknik ini adalah menemukan fungsi pemisah (*classifier*) terbaik diantara fungsi yang tidak terbatas

jumlahnya untuk memisahkan dua atau lebih macam objek. Pemisah berupa garis untuk data yang memiliki dua variabel (dua dimensi), *plane* untuk data yang memiliki tiga variabel (tiga dimensi), dan *hyperplane* untuk data yang memiliki lebih dari tiga variabel.



Gambar 2. 3 Alternatif fungsi pemisah optimal untuk data dua dimensi untuk mencari hyperplane terbaik (Sumber : Gareth dkk, 2008)

Mencari *hyperplane* (pemisah) terbaik selaras dengan mencari *margin* terbesar, yaitu mencari jarak terbesar antara *hyperplane* dengan objek terdekat (*support vectors*). Berikut merupakan persamaan yang digunakan untuk mencari fungsi *hyperplane* terbaik untuk problem primal:

$$min\ \frac{1}{2}\ ||w||^2 \qquad (2.6)$$

Subject to:

$y_i(wx_i + b) \geq 1, i = 1,2, \dots, \ell$ (untuk kelas $y_i = +1$)

$y_i(wx_i + b) \leq -1, i = 1,2, \dots, \ell$ (untuk kelas $y_i = -1$)

Dimana x_i merupakan data masukan dan y_i adalah keluaran dari data x_i. Sedangkan w, b merupakan parameter-parameter yang akan dicari nilainya. Di dalam kasus yang tidak *feasible* dimana beberapa data mungkin tidak dapat dikelompokkan dengan benar, persamaannya menjadi sebagai berikut:

$$min\ \frac{1}{2}\ ||w||^2 + C\sum_{i=1}^{\ell} t_i \qquad (2.7)$$

Subject to:

$y_i(wx_i + b) + t_i \geq 1, i = 1,2, \dots, \ell$ (untuk kelas $y_i = +1$)

$y_i(wx_i + b) + t_i \leq -1, i = 1,2, \ldots, \ell$ (untuk kelas $y_i = -1$)

$t_i \geq 0, \ i = 1,2, \ldots, \ell$

Dimana t_i merupakan variabel *slack.* Tujuan dari persamaan (2.7) adalah untuk meminimalkan kesalahan klasifikasi (*misclassification error*) yang dinyatakan dengan adanya variabel *slack* t_i, dimana dilain sisi pada waktu yang bersamaan persamaan (2.7) berusaha memaksimalkan nilai margin, $\frac{1}{||w||}$. Penggunaan variabel *slack* t_i adalah untuk mengatasi kasus ketidaklayakan (*infeasibility*) dari pembatas (*constraints*) $y_i(wx_i + b) \geq 1$ dengan cara memberi pinalti untuk data yang tidak memenuhi pembatas tersebut. Untuk meminimalkan nilai t_i ini, kita berikan pinalti dengan menerapkan konstanta ongkos C. Sedangkan vektor w tegak lurus terhadap fungsi pemisah: $wx_i + b = 0$ dan konstanta b menentukan lokasi fungsi pemisah relatif terhadap titik asal (*origin*).

Dalam implementasi *machine learning* di lapangan, terdapat data yang tidak sepenuhnya linear sehingga apabila menggunakan dengan pendekatan linear, hasil klasifikasinya menjadi kurang optimal. Sehingga diperlukan sebuah pendekatan untuk memodifikasi data tersebut agar diperoleh *hyperplane* yang baik (Eremenko, 2016). Maka dari itu, *support vector machine* membutuhkan algoritma bantuan dalam melakukan pemisahan (klasifikasi) terhadap data yang sedang diamati. Trik kernel disini berperan dalam membantu untuk melakukan klasifikasi terhadap kejadian tersebut tanpa harus membawanya ke dimensi yang lebih tinggi. Berikut merupakan beberapa trik kernel yang umum diaplikasikan dalam *machine learning*:

a. Exponential Kernel
$$\kappa(x, y) = \exp\left(-\alpha\left||x - y|\right|\right) \qquad \alpha > 0 \qquad (2.7)$$

b. Sigmoid Kernel
$$\kappa(x, y) = \tanh(\alpha x^T y + c) \qquad \alpha > 0, \ c \geq 0 \qquad (2.8)$$

c. Polynomial Kernel
$$\kappa(x, y) = (\alpha x^T y + c)^d \qquad \alpha > 0, \ c \geq 0, d \in \mathrm{Z}_+ \qquad (2.9)$$

d. Gaussian RBF Kernel
$$\kappa(x, y) = \exp\left(\frac{||x - x_i||^2}{2\sigma^2}\right) \qquad (2.10)$$

2.7 Naïve Bayes

Bayes merupakan teknik prediksi berbasi probabilistik sederhana yang berdasar pada penerapan teorema atau aturan *Bayes* dengan asumsi independensi (ketidaktergantungan) yang kuat *(naïve*). Maksud dari independensi yang kuat adalah bahwa sebuah fitur pada sebuah data tidak berkaitan dengan ada atau tidaknya fitur lain pada data yang sama. Berikut merupakan prediksi *Bayes* didasarkan pada teorema *Bayes* dengan formula umum sebagai berikut:

$$P(H|E) = \frac{P(E|H)x\,P(H)}{P(E)} \quad (2.11)$$

Tabel 2. 1 Penjelasan Formula Teorema *Bayes*

Parameter	Keterangan
P (H|E)	Probabilitas akhir bersyarat *(Conditional Probability*) suatu hipotesis H terjadi jika diberikan bukti (*evidence*) E terjadi
P (E|H)	Probabilitas sebuah bukti E terjadi akan mempengaruhi hipotesis H.
P(H)	Probabilitas awal (priori) hipotesis H terjadi tanpa memandang bukti apapun
P(E)	Probabilitas awal (priori) bukti E terjadi tanpa memandang hipotesis/ bukti yang lain

Ide dasar dari aturan *Bayes* adalah bahwa hasil dari hipotesis atau peristiwa (H) dapat diperikirakan berdasarkan pada beberapa bukti (E) yang diamati.

2.7.1 Naïve Bayes untuk Klasifikasi

Klasifikasi *Naïve Bayes* merupakan bagian dari *supervised learning* dengan menggunakan pendekatan metode statistika. Kaitan atara *Naïve Bayes* dengan klasifikasi, korelasi hipotesis dan bukti klasifikasi adalah bahwa hiptesis dalam teorema *Bayes* merupakan label kelas yang menjadi target pemetaan dalam klasifikasi, sedangkan bukti-bukti yang ada merupakan fitur yang berperan sebagai masukan dalam model klasifikasi. Jika X adalah vektor masukan yang berisi fitur dan Y adalah label kelas, *Naïve Bayes* dapat dituliskan dengan P(X|Y). Notasi tersebut memiliki arti bahwa probabilitas lael Y diperoleh setelah melakukan

pengamatan terhadap fitur-fitur yang ada pada label X. Notasi ini disebut juga sebagai probabilitas akhir (*posterior probability*) untuk Y, sedangkan P(Y) disebut juga sebagai probabilitas awal (*prior proability*).

Selama proses *training* atau latihan, harus dilakukan pembelajaran terhadiap probabilitas akhir (*posterior probability*) pada model untuk setiap kombinasi X dan Y berdasarkan informasi yang diperoleh dari data tes atau data latih. Dengan membangun model tersebut, suatu uji data X' dapat diklasifikasikan dengan mencari nilai Y' dengan memaksimalkan nilai P(X'|Y') yang didapat.

Formulasi *Naïve Bayes* untuk klasifikasi sebagai berikut:

$$P(X|Y) = \frac{P(Y)\prod_{i=1}^{q} P(X_i|Y)}{P(X)} \quad (2.12)$$

Dimana P(X|Y) merupakan probabilitas data dengan vektor X pada kelas Y. P(Y) adalah probabilitas awal kelas Y. $\prod_{i=1}^{q} P(X_i|Y)$ adalah probabilitas independen kelas Y dari semua fitur dalam vektor X. Nilai P(X) selalu tetap sehingga dalam perhitungan prediksi selanjutnya, hanya menghitung bagian pembilang $P(Y)\prod_{i=1}^{q} P(X_i|Y)$ dengan memilih yang terbesar sebagai kelas yang akan dipilih untuk hasil prediksi. Sementara probabilitas independen $\prod_{i=1}^{q} P(X_i|Y)$ tersebut merupakan pengaruh semau fitur dari data terhadap setiap kelas Y, yang dinotasikan dengan:

$$P(X|Y = y) = \prod_{i=1}^{q} P(X_i|Y) = y \quad (2.13)$$

Dimana setiap set fitur X = {X_1, X_2, X_3, …, X_q} terdiri atas *q* atribut (*q* dimensi).

2.8 Kinerja Metode

Dalam mengetahui kinerja suatu model *learning* yang telah dibuat, diperlukan pengukuran terhadap kinerja. Dalam kasus *supervised learning*, pendekatan yang dapat dilakukan adalah dengan membandingkan data yang telah dilakukan *training* dengan data *ground truth label* atau data *output*-nya memiliki label. Sedangkan kasus *unsupervised learning*, pendekatan yang dapat dilakukan adalah dengan menggunakan pendapat para *expert*(orang yang ahli/ berpengalaman di bidang tersebut) untuk mengevaluasi hasil yang diperoleh dari model *learning* dan membandingkan dengan data yang sudah ada dan berkaitan (Santosa, 2018). Klasifikasi tergolong dalam *supervised learning* karena memiliki *ground truth lable*

yang digunakan sebagai pembanding dalam mengukur model *learning* yang telah dibuat. Berikut merupakan metode pengukuran kinerja untuk model *learning* klasifikasi:

Tabel 2. 2 Ilustrasi dalam mengukur peformansi model dengan *confusion matrix*

N	Aktual = - 1	Aktual = +1	
Prediksi = -1	TN (*True Negative*) (a)	FP (*False Negative*) (c)	(a) + (c)
Prediksi = +1	FP (*False Poistive*) (b)	TP (*True Positive*) (d)	(b) + (d)
	(a) + (b)	(c) + (d)	

Berikut merupakan keterangan untuk masing-masing isi tabel:

a. TN (*True Negatif*) = *Output* kelas negatif yang berhasil diprediksi sebagai kelas negatif.
b. FP (*False Positive*) = *Output* kelas negatif yang berhasil diprediksi sebagai kelas positif.
c. FN (*False Negative*) = *Output* kelas positif yang berhasil diprediksi sebagai kelas negatif.
d. TP (*True Positive*) = *Output* kelas positif yang berhasil diprediksi sebagai kelas positif.

Dari keempat keterang yang diperoleh dari *confusion matrix* diatas, berikut merupakan matrik evaluasi yang dapat digunakan:

a. $\text{Akurasi} = \frac{(\text{TN+TP})}{(\text{jumlah data})} = \frac{(a)+(d)}{(a)+(b)+(c)+(d)}$ (2.14)

b. $\textit{Miss classification rate} = \frac{(\text{FP+FN})}{(\text{jumlah data})} = \frac{(b)+(c)}{(a)+(b)+(c)+(d)}$ (2.15)

c. $\textit{Recall/ Sensitivity/ True Positive Rate} = \frac{\text{TP}}{\text{Aktual “+1”}} = \frac{(a)}{(c)+(d)}$ (2.16)

d. $\text{Presisi} = \frac{\text{TP}}{\text{Prediksi “ + 1”}} = \frac{(d)}{(b)+(d)}$ (2.17)

e. $\textit{False Potive Rate/ False Alarm Rate} = \frac{\text{FP}}{\text{Aktual " − 1"}} = \frac{(b)}{(a)+(b)}$ (2.18)

f. *Specifity* = $\frac{\text{TN}}{\text{Aktual "}-1\text{"}} = \frac{(a)}{(a)+(b)}$ (2.19)

2.9 Penelitian Terdahulu

Penelitian mengenai prediksi pelanggan *churn* pada industri telekomunikasi telah banyak dilakukan. Secara garis besar penelitian yang dilakukan bersifat kuantitatif, namun topik yang diangkat memiliki bahasan yang berbeda-beda. Topik yang diangkat diantaranya mengenai faktor yang berpengaruh terhadap terjadinya pelanggan *churn*, penerapan salah satu metode yang ada pada *data mining* untuk prediksi pelanggan *churn*, hingga komparasi metode yang ada pada *data mining* untuk memilih metode yang paling baik dalam memprediksi terjadinya pelanggan *churn*.

Penelitian mengenai analisis faktor yang berpengaruh terhadap terjadinya pelanggan *churn* pernah dilaksanakan oleh Ahn, Jae-Hyeon et.al (2006) dalam penelitian *Customer churn analysis*: *Churn determinants and mediation effects of partial defection in the Korean mobile telecommunications service industry*. Didalam penelitian tersebut, peneliti melakukan analisis terhadap beberapa variabel yang terjadinya pelanggan *churn* pada suatu industri telekomunikasi. Dari keseluruhan variabel yang dianalisa, terdapat beberapa variabel yang dinilai signifikan pengaruhnya terhadap terjadinya pelanggan *churn* yang diantaranya: frekuensi keluhan pelanggan yang diterima perusahaan, tingkat loyal, hingga jumlah tagihan.

Pada penelitian terkait implementasi metode yang terdapat pada *data mining*, telah dilakukan oleh Hanifa, Tesha Tasmalaila et.al (2017) dalam penelitian Analisis Churn Prediction pada Data Pelanggan PT. Telekomunikasi dengan *Logistic Regression* dan *Underbagging*. Penelitian tersebut bertujuan untuk membangun model prediksi dengan menggunakan *logistic regression* pada kasus *imbalance data*. Peneliti memperoleh data yang tidak seimbang antara *churn* (7.513 *record*) dan *non-churn* (192.848 *record*) sehingga peneliti menilai diperlukan metode *Underbagging* untuk penanganan kelas tidak seimbang dengan teknik sampling. Performansi yang dihasilkan dari penelitian ini memiliki tingkat akurasi yang cukup baik yaitu sebesar 85.53%.

Penelitian lainnya yang berkaitan dengan implementasi salah satu metode yang terdapat dalam *data mining* dilakukan oleh Suryana, Nana (2012) dalam penelitiannya Prediksi Churn Dan Segmentasi Pelanggan TV Berlangganan untuk studi kasus Transvision di Jawa Barat. Penelitian tersebut bertujuan untuk membangun sebuah model prediksi *churn* sebelum kemudian melakukan klastering agar diperoleh informasi dari pelanggan yang mempunyai peluang *churn* tinggi sehingga dapat disusun rencana strategis serta jenis promosi dari perusahaan berdasarkan segementasinya. Melalui pendekatan metode *decision tree* dan *k-means clustering* diperoleh tingkat akurasi model sebesar 90.89% dengan total 5792 pelanggan yang mendapatkan penawaran program retensi.

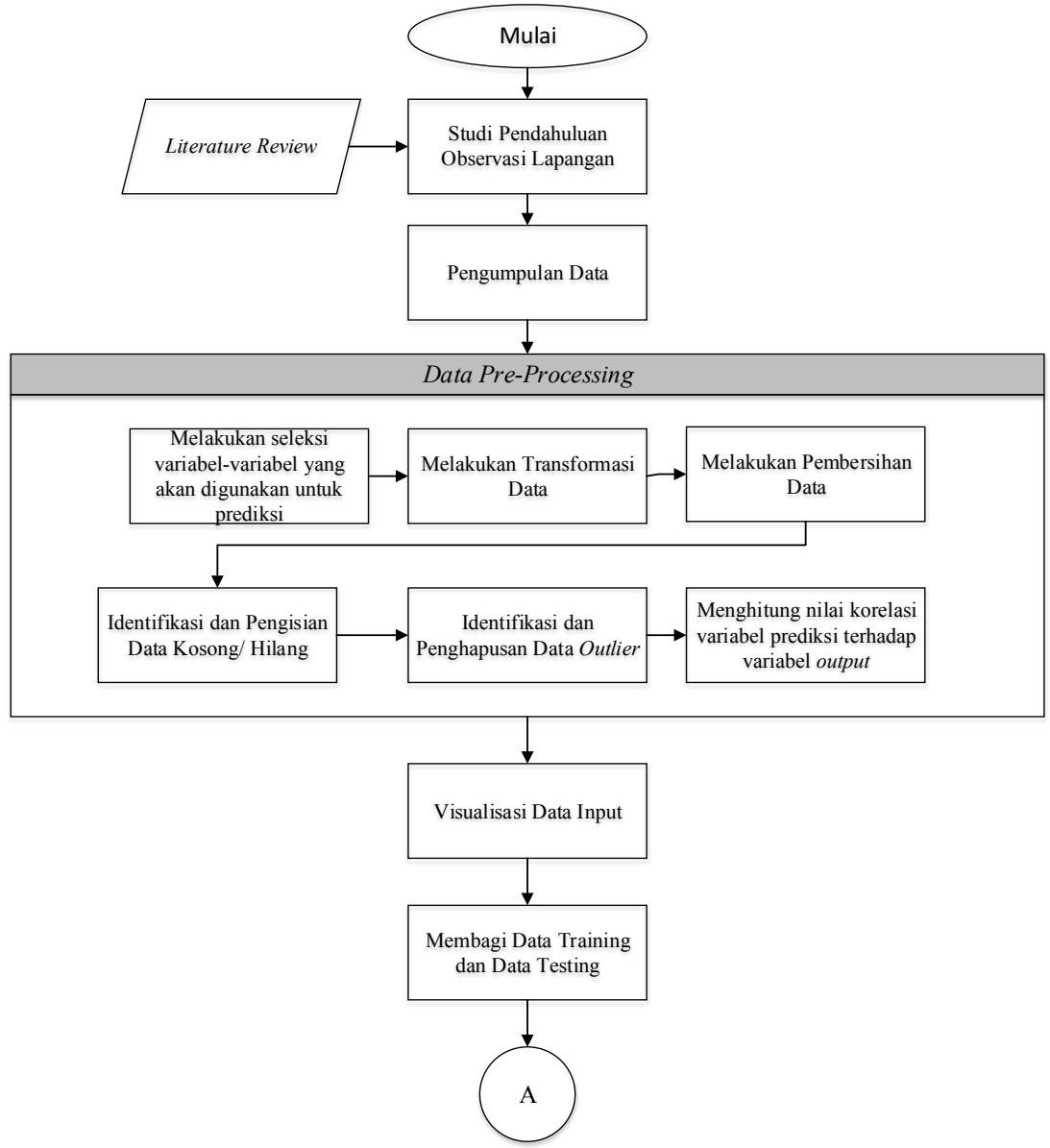
Selain itu, penetlitian yang melakukan pendekatan dengan beberapa metode sekaligus dalam upaya memilih metode yang paling optimal dalam melakukan prediksi pelanggan *churn*, pernah dilakukan oleh Madavossi, Zakki (2009) dalam penelitian Aplikasi *Data Mining* Untuk *Churn Prediction*. Penelitian tersebut bertujuan untuk membangun model prediksi *churn* dengan komparasi metode *decision tree* dan *artificial neural network* dengan objek amatan pelanggan Flexy Classy. Hasil penelitian menunjukkan bahwa model yang dibangun dengan pendekatan *decision tree* memiliki tingkat performansi yang paling baik dengan *overall accuracy* sebesar 94.73%. Sedangkan, model yang dibangun dengan pendekatan *artificial neural network* memiliki *overall accuracy* sebesar 88.36%.

Tabel 2. 3 Penelitian-Penelitian Terdahulu

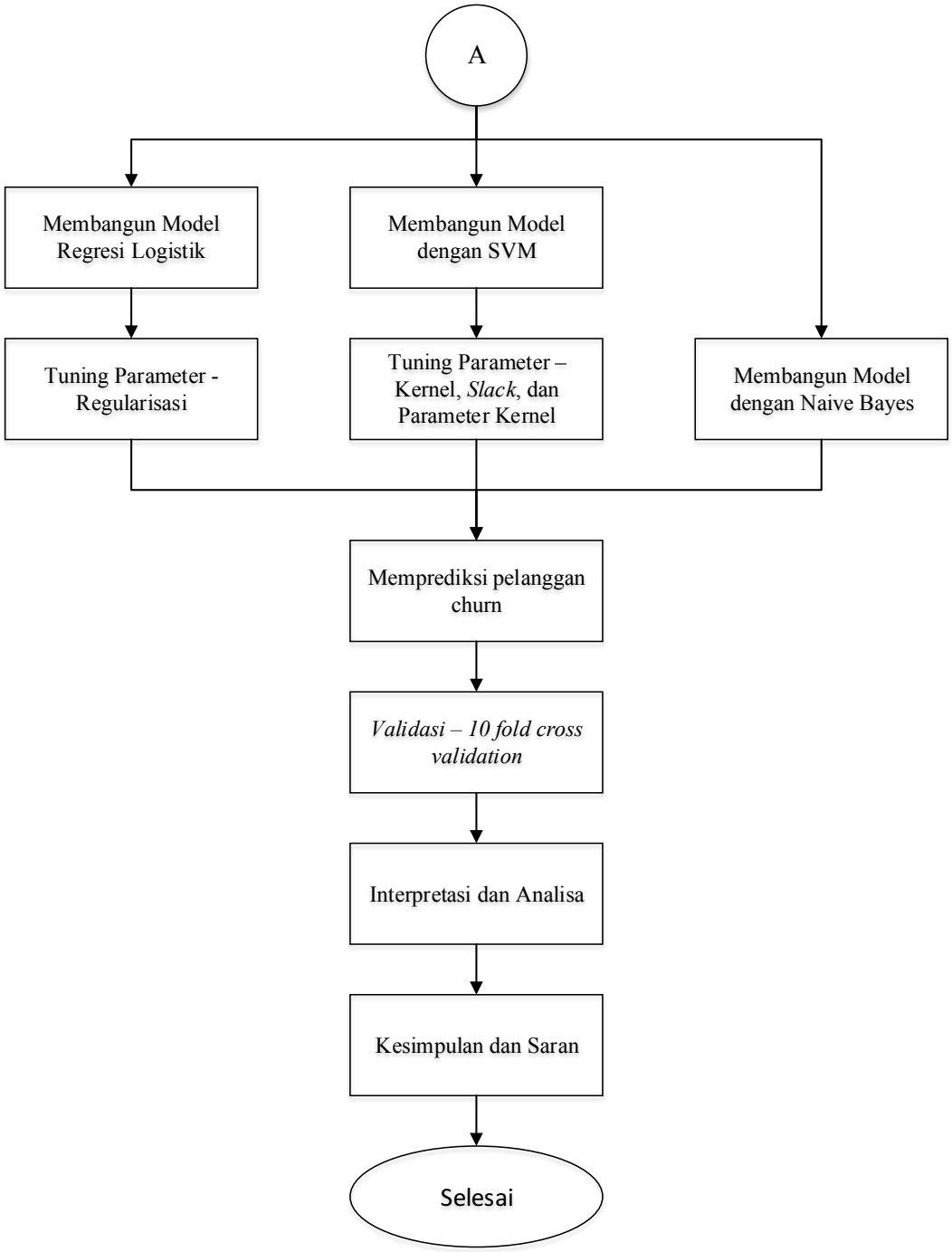
Judul	Penulis	Tahun	Metode	Output
Customer churn analysis: *Churn determinants and mediation effects of partial defection in the Korean mobile telecommunications service industry*.	Jae-Hyeon Ahn, Sang-Pil Han, & Yung-Seop Lee	2006	Analisis deskriptif	Deskripsi dan analisis mengenai faktor-faktor apa saja yang dinilai berpengaruh secara signifikan terhadap terjadinya *churn* pada pelanggan industri telekomunikasi.
Analisis Churn Prediction pada Data Pelanggan PT. Telekomunikasi dengan Logistic Regression dan Underbagging	Tesha Tasmalaila Hanifa, Adiwijaya, & Said Al-Faraby	2017	Regresi Logistik & Underbagging	Data mentah yang *imbalance* sehingga diperlakuan perlakuan khusus serta model prediksi *churn* dengan tingkat *overall accuracy* sebesar 85.53%
Prediksi Churn dan Segmentasi Pelanggan TV Berlangganan (Studi Kasus Transvision Jawa Barat)	Nana Suryana, ST., M.Kom	2012	Decision Tree & K-Means Clustering	Model prediksi *churn* dengan tingkat *overall accuracy* sebesar 90.89% serta pembagian klaster yang dapat menjadi referensi bagi perusahaan dalam memperoleh promo atau penawaran retensi khusus
Aplikasi Data Mining Untuk Churn Prediction	Zakki Madavossi	2009	Decision Tree & Artficial Neural Network	Model prediksi *churn* dengan tingkat *overall accuracy* yang lebih baik diantara 2 metode (*decision tree* dan *artificial neural network*). Diperoleh metode *decision tree* lebih baik dengan tingkat *overall accuracy* sebesar 94.73%

BAB 3
METODOLOGI PENELITIAN

Pada Bab ini akan dijelaskan mengenai metodologi yang digunakan untuk menyelesaikan penelitian. Secara garis besar, metodologi dan tahapan pengerjaan dijelaskan melalui *flowchart* dibawah ini.



Gambar 3. 1 *Flowchart* Pengerjaan Penelitian



Gambar 3. 2 *Flowchart* Pengerjaan Penelitian (lanjutan)

3.1 Studi Pendahuluan

Studi pendahuluan diawali dengan mengidentifikasi bidang yang akan digunakan dalam penelitian tugas akhir sebelum kemudian menentukan topik penelitian tugas akhir berupa komparasi metode dalam prediksi pelanggan *churn* dengan objek studi pelanggan PT X di kawasan Bogor. Pemahaman mengenai

prediksi pelanggan *churn* dilakukan dengan mempelajari beberapa sumber mulai dari internet, buku, hingga penelitian terdahulu yang berhubungan secara langsung maupun tidak langsung terhadap *data mining,* metode regresi logistik, *support vector machine*, *naïve bayes*, hingga *churn prediction* itu sendiri. Dari studi literatur tersebut diperoleh beberapa saran berupa metode yang digunakan dalam melakukan *churn prediction* yang diantaranya regresi logistik, *support vector machine*, serta *naïve bayes*.

3.2 Pengumpulan Data

Pengumpulan data yang dilakukan menggunakan data informasi pengguna jasa layanan PT X di area kerja Bogor. Data yang dibutuhkan untuk pengembangan model prediksi *churn* merupakan sebuah *dataset* yang terdiri dari sebuah atribut respon (*dependent variabel*) dan (*independent variabel*). *Dependent variable* yang dimaksud merupakan atribut respon yang berisi keterangan pelanggan *churn* atau *non-churn* sedangkan untuk *independent variable* sendiri terdiri dari beberapa atribut prediktor yang diantaranya yaitu: wilayah asal, keterangan pengeluaran pelanggan untuk jasa *voice,* SMS, *data*, *digital*, dan jumlah penggunaan *byte* untuk data. Terkait *dependent variable*, diperoleh dari pelanggan dengan status A (*active*) untuk pelanggan yang dikategorikan *non-churn* dan status I (*inactive*) dan G (*grace*) untuk pelanngan yang dikategorikan sebagai *churn*.

3.3 *Data Pre-Processing*

Setelah data yang dibutuhkan diperoleh, selanjutnya adalah dengan melakukan *data pre-processing*. Pertama adalah dengan melakukan pemilihan variabel yang digunakan dari data yang diperoleh. Selanjutnya adalah melakukan transformasi data yang belum berjenis numerik. Hal ini mengingat aplikasi *Anaconda* cenderung hanya bisa mengolah data dalam bentuk numerik. Kemudian dilakukan pengisian data yang kosong atau hilang dengan rata-rata masing-masing variabel. Selanjutnya, setelah dilakukan pengisian data yang kosong, dilakukan identifikasi dan penghilangan data *outlier.* Kemudian dilakukan penentuan signifikansi dan *multicollinearity check* untuk mengetahui apakah ada variabel yang memiliki relasi cukup tinggi (lebih dari 0.8). Apabila terdapat dua ata lebih variabel

yang memiliki hubungan cukup tinggi, salah satu variabel dihapus atau tidak dilibatkan dalam pembuatan model *classifier.*

3.4 *Data Processing*

Selanjutnya dilakukan visualisasi terhadap data yang telah dilakukan *data pre-processing* untuk mengetahui karakteristik pelanggan berdasarkan masing-masing variabel. Selanjutnya, dilakukan pembagian dataset menjadi data *trainng* dan data *testing* sebelum kemudian dilakukan pembuatan model *classifier* dengan metode regresi logistik, *support vector machine,* dan *naïve bayes.*

3.5 Pembangunan dan Analisa Model

Setelah dilakukan prediksi pelanggan *churn*, selanjutnya dilakukan pembangunan dan interpretasi model. Pada metode regresi logistik, dilakukan satu *tuning parameter* yaitu jenis regularisasinya apakah menggunakan L1 atau L2. Pada metode *support vector machine* dilakukan 1 *tuning parameter* untuk kernel linear (nilai *slack*), 2 *tuning parameter* untuk kernel RBF (nilai *slack* dan gamma), dan 2 *tuning parameter* untuk kernel Polinomial (nilai *slack* dan *degree*). Pada metode *naive bayes* hanya dibangun satu model atau tidak dilakukan *tuning parameter*. Selanjutnya, masing-masing metode dilakukan prediksi pelanggan *churn* terhadap data *testing* dan dilakukan validasi model dengan *10-fold cross validation.* Terakhir dilakukan rekapitulasi rata-rata performansi masing-masing model yang terdiri diantaranya: *accuracy, recall, precision, f1-score,* dan *missclassification rate*.

Terkait interpretasi dari hasil performansi masing-masing model dalam mengklasifikasikan pelanggan *churn* dan *non-churn*. Selanjutnya dilakukan analisa terkait kinerja dari masing-masing model yang digunakan dalam prediksi pelanggan *churn.*

3.6 Kesimpulan dan Saran.

Setelah komparasi antar metode dilakukan, penarikan kesimpulan dilakukan terkait dengan tujuan yang ingin diraih sekaligus untuk memperjelas proses dan hasil penelitian yang telah dilakukan. Kemudian, diberikan rekomendasi model yang paling baik kepada PT X serta saran bagi penelitian selanjutnya.

BAB 4
PENGUMPULAN DAN PENGOLAHAN DATA

Pada BAB ini akan ditampilkan mengenai proses pengumpulan dan pengolahan data secara sistematis sesuai kerangka pemikiran yang telah dibuat.

4.1 Pengumpulan Data

Penelitian ini akan menggunakan data pelanggan PT X dikawasan Bogor dan sekitarnya. Data awal sebelum dilakukan seleksi berjumlah 122.570 data dan setelah dilakukan seleksi terhadap data mentah tersebut diperoleh 8.173 data dengan komposisi 3.973 data dengan kategori *churn* dan 4200 data dengan kategori *non-churn*. Seleksi yang dilakukan berupa pemilihan data pada variabel "*Branch*" terbatas pada kawasan Bogor, kemudian dilakukan seleksi terhadap variabel "Brand" terhadap pada salah satu produk dari PT X.

Kerangka waktu untuk data pelanggan adalah dari Januari 2017 hingga Juni 2018. Dalam penelitian ini, pelanggan yang memiliki parameter *churn* atau berhenti menggunakan layanan dari PT X pelanggan yang dikategorikan sebagai *churn* atau secara sukarela berhenti menggunakan layanan PT X memiliki parameter status "G" (*Grace,* masa tenggang) dengan rentang masa tenggang waktu lebih dari 30 hari atau "I" (*Inactive*, tidak aktif). Sedangkan pelanggan yang dikategorikan sebagai *non-churn* atau pelanggan yang masih menggunakan layanan dari PT X memiliki parameter status "A" (*Active*, masa aktif) atau status "G" dengan rentang waktu masa tenggang kurang dari 30 hari. Diperoleh 11 jenis variabel seperti yang dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Variabel-Variabel Hasil Pengumpulan Data

No	Variabel	No	Variabel
1	Region	7	SMS
2	Branch	8	Data
3	Cluster	9	Digital
4	Kecamatan	10	Payload
5	Brand	11	Status
6	Voice		

4.2 Variabel *Input* dan *Output*

Variabel yang dijadikan sebagai variabel input atau *independent variable* untuk proses *training* dan *testing* merupakan faktor-faktor yang diperoleh dari kegiatan pengumpulan data untuk membangun model prediksi pelanggan *churn*er *churn*.

Variabel 1 berisi data terkait area kerja PT X yaitu *eastern jabodatebek*. Variabel 2 berisi data terkait percabangan dari variabel 1 yang terbagi menjadi branch Bogor dan branch Karawang. Peneliti melakukan seleksi terhadap data dengan hanya menggunakan branch pada branch Bogor. Kemudian untuk variabel 3 yang berisi terkait wilayah yang dicakup dari Branch Bogor yang terdiri dari Bogor, Cibubur, dan Depok yang selanjutnya dikategorikan sebagai variabel *input* X1. Selanjutnya untuk variabel 4, berisi kecamatan-kecamatan yang tercakup dalam ketiga kabupaten yang terdapat pada variabel 3. Variabel 5 berisi jenis *brand* yang diatawarkan oleh PT X di area kerja *eastern jabodetabek.* Peneliti melakukan seleksi kembali terhadap variabel 5 dengan hanya memilih salah satu *brand* dari keseluruhan *brand* yang ditawarkan PT X di area kerja tersebut.

Untuk variabel 6 berisi informasi terkait rata-rata jumlah pengeluaran pelanggan pengguna layanan PT X untuk jenis layanan panggilan suara dan dikategorikan variabel *input* X2. Variabel 7 berisi informasi terkait rata-rata jumlah pengeluaran pelanggan pengguna layanan PT X untuk jenis layanan SMS (*short massage service*) dan dikategorikan variabel *input* X3. Selanjutnya, pada variabel 8 berisi informasi terkait rata-rata jumlah pengeluaran pelanggan pengguna layanan PT X untuk jenis layanan akses paket data internet dan dikategorikan variabel *input* X4. Variabel 9 berisi informasi terkait rata-rata jumlah pengeluaran pelanggan pengguna layanan PT X untuk jenis layanan tambahan yang disediakan oleh PT X dan dikategorikan variabel *input* X5. Variabel 10 berisi informasi terkait rata-rata jumlah penggunaan data dalam satuan *byte* dan dikategorikan variabel *input* X6. Terakhir, variabel 11 berisi informasi terkait keterangan pelanggan apakah aktif (A), memasuki masa tenggang (G), tidak aktif (I), atau tidak diketahui statusnya (?) dikategorikan sebagai variabel *output*.

Tabel 4. 2 Variabel *Input*

No	Variabel	No	Variabel
X1	Cluster	X4	Data
X2	Voice	X5	Digital
X3	SMS	X6	Payload

Data variabel *input* X1 memiliki tipe data kategorikal, sedangkan X2, X3, X4, dan X5 memiliki tipe data numerik. Data variabel *output* memiliki tipe data kategorikal.

4.3 Data Cleaning

Pada bagian ini dilakukan pembersihan terhadap data yang telah dilakukan seleksi. Pada bagian ini terdapat beberapa pengolahan yang dilakukan terhadap data yang diantaranya yaitu: pengecekan dan konversi bentuk data (*labeling*), pengisian data yang kosong, identifikasi dan menghilangkan data yang *outlier*, menentukan signifikansi masing-masing *independent* variable atau variabel *input,* dan *multicollinearity check.*

4.3.1 Pengecekan dan Konversi Bentuk Data (*Labeling)*

Pertama, dilakukan pengecekan isi data apakah sepenuhnya telah memiliki tipe data yang serupa. Hasilnya ditemukan bahwa variabel X1 memiliki tipe data *string*, sedangkan X2, X3, X4, dan X5 memiliki tipe data *float*. Hal ini juga mengindikasikan bahwa data sepenuhnya telah bebentuk numerik. Hal ini membuat X1 perlu untuk dilakukan konversi terhadap variabel X1 dari bentuk data yang sebelumnya *string* ke bentuk data *numerik* dengan pemberian label. Pemberian label dilakukan dengan merubah isi yang terdapat pada variabel *cluster* menjadi sebuah nilai. Berikut merupakan keadaan variabel sebelum dan setelah diberikan label pada variabel cluster:

Tabel 4. 3 Konversi Data Variabel *Cluster*

Data Awal	Data Akhir
CIBUBUR	1
DEPOK	2
BOGOR	3

Selain itu, *dependent variable* juga dilakukan pemberian label terhadap data yang sebelumnya memiliki nilai A atau *active* menjadi 0 dan data yang sebelumnya memiliki nilai I (*inactive*) atau G (*Grace*) menjadi 1.

4.3.2 Pengisian Data yang Kosong/ Hilang

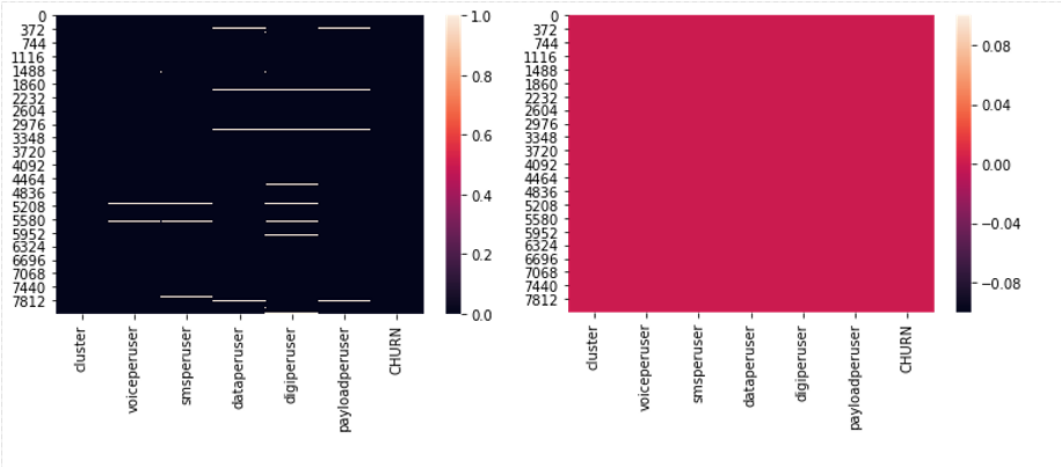
Selanjutnya adalah identifikasi terhadap ketersediaan data secara keseluruhan apakah telah sepenuhnya terisi atau masih terdapat bagian yang belum terisi. Hasilnya diperoleh beberapa bagian data yang kosong pada beberapa variabel. Identifikasi data yang kosong dilakukan dengan bantuan *library* yang tersedia pada aplikasi *Spyder*, yaitu *seaborn*. *Seaborn* sendiri merupakan salah satu fitur bawaan yang berfungsi dalam memvisualisasikan gambar pada aplikasi *Anaconda Navigator, environment* Spyder, bahasa pemrograman *Python.* Pada aplikasinya, *seaborn* dapat memvisualisasikan lokasi data yang hilang dengan memberikan keterangan informasi berupa perbedaan warna (Gambar 4.1).

Salah satu langkah untuk menghadapi hal tersebut adalah dengan mengisi data yang kosong/ tidak memiliki nilai dengan *mean* dari masing-masing variabel(Santosa, 2018). Berikut merupakan jumlah masing-masing variabel prediktor yang memiliki tipe data *integer* dan kosong/ hilang.

Tabel 4. 4 Jumlah Data Kosong dan Rata-Rata Masing-Masing Variabel

Variabel	Jumlah Data Kosong	Rata-Rata
Voice	127	18.170,85
SMS	105	6.666,27
Data	220	34.778,77
Digital	337	4.661,50
Payload	220	6.324.491.382,43

Pada Tabel 4.4 diketahui jumlah data yang kosong beserta rata-rata dari masing-masing variabel. Dari seluruh data yang teridentifikasi kosong, selanjutnya dapat dilakukan pengisian data yang kosong dengan rata-rata dari masing-masing variabel. Kemudian dilakukan pengecekan kembali dengan *library seaborn* untuk memastikan sudah tidak terdapat data yang kosong (Gambar 4.1).



Gambar 4. 1 Data sebelum (sebelah kiri) dan setelah (sebelah kanan) dilakukan pengisian data yang hilang dengan bantuan *library seaborn*.

4.3.3 Outlier Identification

Setelah dilakukan pengisian terhadap data yang kosong/ tidak memiliki nilai, selanjutnya *outlier identification* untuk memastikan apakah terdapat data yang *outlier* atau tidak. Pengecekan dilakukan dengan menggunakan metode Tukey IQR. Nilai yang dikategorikan sebagai *outlier* merupakan nilai yang berada dibawah $Q_1 - 1.5\,IQR$ atau nilai yang berada di atas $Q_3 + 1.5\,IQR$ dengan Q_1dan Q_3 merupakan kuartil bawah dan kuartil atas, sedangkan IQR (*Inter Quartile Range*) merupakan jarak antar kuartil. Diperoleh informasi sebelum dilakukan identifikasi dan penghilangan data yang *outlier* dengan penerapan Tukey IQR untuk masing-masing variabel input sebagai berikut. Tabel 4.5 diketahui informasi terhadap data sebelum dilakukan *outlier identification*.

Tabel 4. 5 Informasi data masing-masing variabel sebelum dilakukan pembersihan data *outlier*

Variabel	Mean	Standard Deviation	Median	Q1	Q3	IQR	Jumlah Outlier
Voice	18.170,83	14.238,24	15.063	5.718	28.571	22.853	12
SMS	6.666,24	4.381,90	5.602	2.806	10.378	7.252	4
Data	34.778,75	29.936,54	36.674	4.458	61.944	57.486	1
Digital	4.661,48	4.904,53	2.956	1.179	7.302	6.123	159
Payload	6,32e+9	1,35e+10	2,08e+09	4,78e+07	4,99e+09	4.94e+09	1020

Dari keadaan diatas dilakukan penghilangan terhadap data *outlier*. Dibutuhkan sebanyak 16 kali iterasi agar masing-masing variabel tidak memiliki data yang *outlier*. Pada Tabel 4.6 diperoleh hasil penghilangan data *outlier* melalui pendekatan Tukey IQR. Diperoleh hasil reduksi jumlah data yang sebelumnya berjumlah 8173 data menjadi 6788 data. Berikut merupakan informasi masing-masing variabel setelah dilakukan pembersihan data *outlier*.

Tabel 4. 6 Informasi data masing-masing variabel setelah dilakukan terhadap pembersihan data *outlier*

Variabel	Mean	Standar Deviasi	Median	Q1	Q3	IQR	Jumlah Outlier
Voice	18.365,04	14.493,39	13793	5926,5	29938	24011,5	0
SMS	6759,27	4483,59	5419	2867	10768	7901	0
Data	35123,48	30224,29	37074	4404	63024	58620	0
Digital	4343,40	3838,60	2862	1243,5	7175	5931.5	0
Payload	2,03e+10	1,35e+10	1,33e+9	3,14e+7	3,57e+9	3,54e+9	0

4.3.4 Penentuan Variabel yang Dilibatkan dalam Pembangunan Model

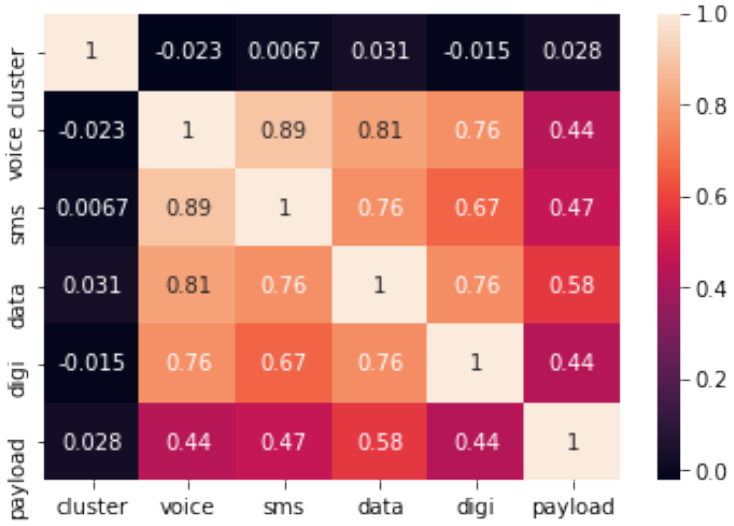
Selanjutnya, dilakukan perhitungan dengan salah satu metode yang digunakan untuk mengestimasi fungsi regresi yaitu *logit* dengan data yang telah dilakukan normalisasi terlebih dahulu. Hal tersebut bertujuan untuk mengetahui signifikansi dari masing-masing variabel terhadap variabel respon atau *dependent variable*. Perhitungan dilakukan pada aplikasi Anaconda Navigator *environment* Python dengan melakukan *import library* dari *statsmodes.api*. Dari perhitungan tersebut, diperoleh hasil seperti yang ditampilkan pada Gambar 4.2. Dari gambar tersebut diperoleh informasi mengenai koefisien masing-masing variabel *input* terhadap variabel *output*. Untuk signifikansi, diperlukan normalisasi terlebih dahulu terhadap data *input* agar antar koefisien dari masing-masing variabel bisa dibandingkan. Penjelasan terkait penentuan variabel yang dilibatkan dalam pembuatan model akan dijelaskan lebih lanjut pada sub bab 5.1.

```
Optimization terminated successfully.
         Current function value: 0.057654
         Iterations 11
                         Results: Logit
================================================================
Model:              Logit            Pseudo R-squared: 0.917
Dependent Variable: CHURN            AIC:              794.7062
Date:               2019-01-06 08:51 BIC:              835.6437
No. Observations:   6788             Log-Likelihood:   -391.35
Df Model:           5                LL-Null:          -4704.7
Df Residuals:       6782             LLR p-value:      0.0000
Converged:          1.0000           Scale:            1.0000
No. Iterations:     11.0000
----------------------------------------------------------------
             Coef.   Std.Err.      z      P>|z|     [0.025   0.975]
----------------------------------------------------------------
cluster      3.7664    0.2013   18.7062  0.0000    3.3718   4.1610
voice       -0.0001    0.0000   -5.3941  0.0000   -0.0002  -0.0001
sms          0.0000    0.0001    0.0710  0.9434   -0.0001   0.0001
data        -0.0004    0.0000  -19.3432  0.0000   -0.0004  -0.0003
digi         0.0004    0.0001    7.6222  0.0000    0.0003   0.0005
payload      0.0000    0.0000   12.3456  0.0000    0.0000   0.0000
================================================================
```

Gambar 4. 2 Hasil Perhitungan dengan *Logit Model* terhadap *Independent Variable* dan *Dependent Variable*

4.3.5 Multicollinearity Check

Terakhir dilakukan pengecekan terhadap hubungan antara masing-masing variabel *input*. Hal ini bertujuan untuk mengetahui apakah terdapat variabel yang memiliki multikolinearitas atau tidak. Setelah dilakukan pengecekan, diketahui bahwa terdapat variabel yang memiliki hubungan atau korelasi yang cukup tinggi atau multikolinearitas (lebih dari 0.8). Hal ini diperlukan untuk menghilangkan salah satu dari variabel tersebut untuk mencegah *noise* atau *overfitting* terhadap model. Berdasarkan Gambar 4.2, diketahui bahwa variabel *voice* memiliki korelasi yang cukup tinggi dengan variabel *sms* dan *data*. Dari informasi tersebut, diputuskan untuk tidak menyertakan variabel *voice* ddalam pembuatan model mengingat variabel ini memiliki nilai *p-value* > 0.05 dengan tingkat *confidence level* sebesar 95% serta berdasar Gambar 4.3, variabel ini memiliki hubungan yang cukup kuat dengan variabel *sms* dan *data.*



Gambar 4. 3 Nilai Korelasi antar Variabel *Input*

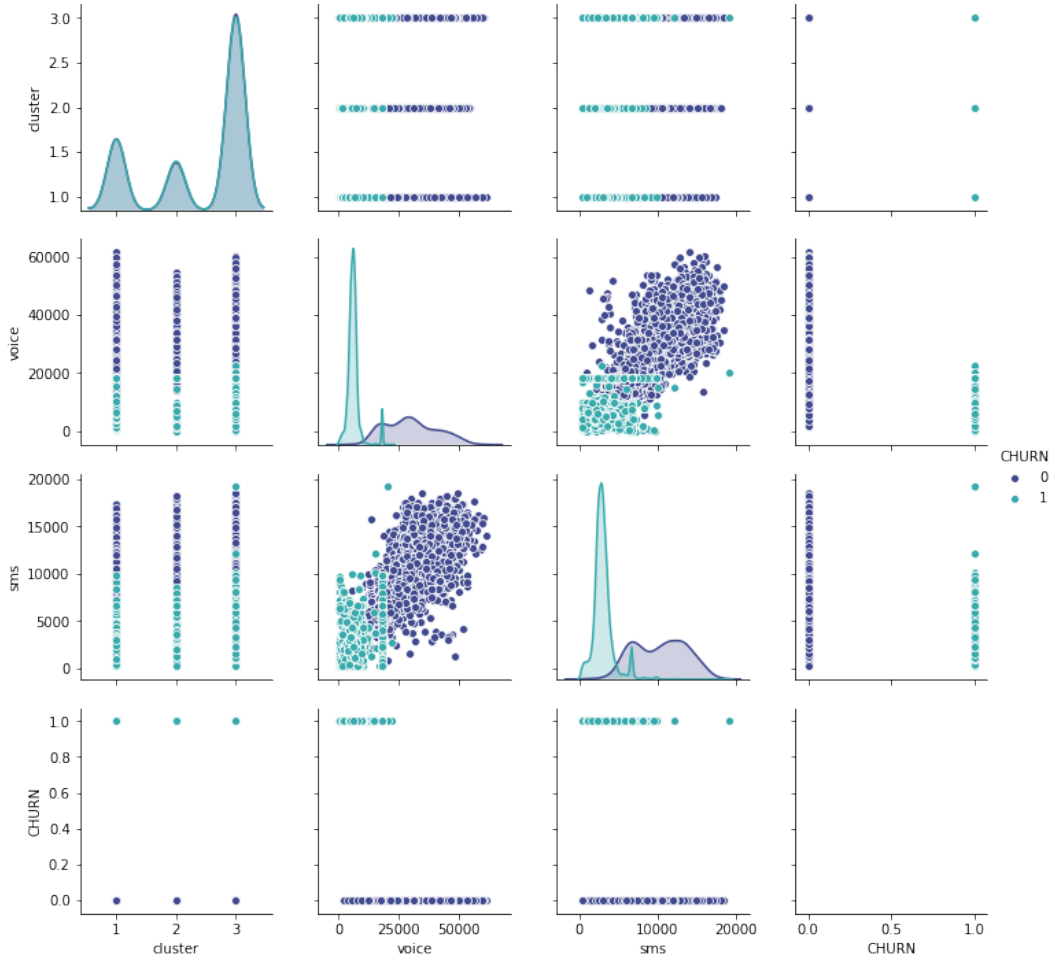
4.4 Data Training dan Data Testing

Data yang telah melalui tahap *preprocessing* data kemudian dibagi menjadi dua bagian sebagai *data training* dan *data testing* dengan porsi 75%:25%. Metode-metode yang digunakan dalam prediksi pelanggan merupakan *supervised learning* sehingga data dilakukan *training* terlebih dahulu yang berguna untuk mengenali pola dari data *input* sehingga dihasilkan sebuah model *classifier* yang dapat digunakan untuk prediksi selanjutnya. Dari hasil partisi diatas diperoleh data *training* berjumlah 5091 data dan data *testing* berjumlah 1697 data.

Setelah dilakukan partisi terhadap data menjadi data *training* dan data *testing*, langkah selanjutnya adalah melakukan normalisasi terhadap data *training*. Hal ini selain bertujuan selain untuk mempercepat proses *running* dalam pembangunan model, juga berperan agar setiap variabel memiliki peran yang sama terhadap model yang dibangun. Kemudian, data yang telah dilakukan normalisasi digunakan untuk pembuatan model *classifier* dengan pendekatan metode regresi logistik, *support vector machine,* dan *naive bayes.* Penerapan masing-masing metode akan dijelaskan di sub bab selanjutnya. Setelah dibangun model dari masing-masing metode, langkah selanjutnya adalah melakukan *k-fold cross validation*. Peneliti menggunakan $k = 10$ dimana hal ini berarti data dibagi menjadi 10 bagian dimana setiap bagian akan memperoleh peran sebagai *data training* maupun *data testing*.

4.5 Statistik Deskriptif *Data Input*

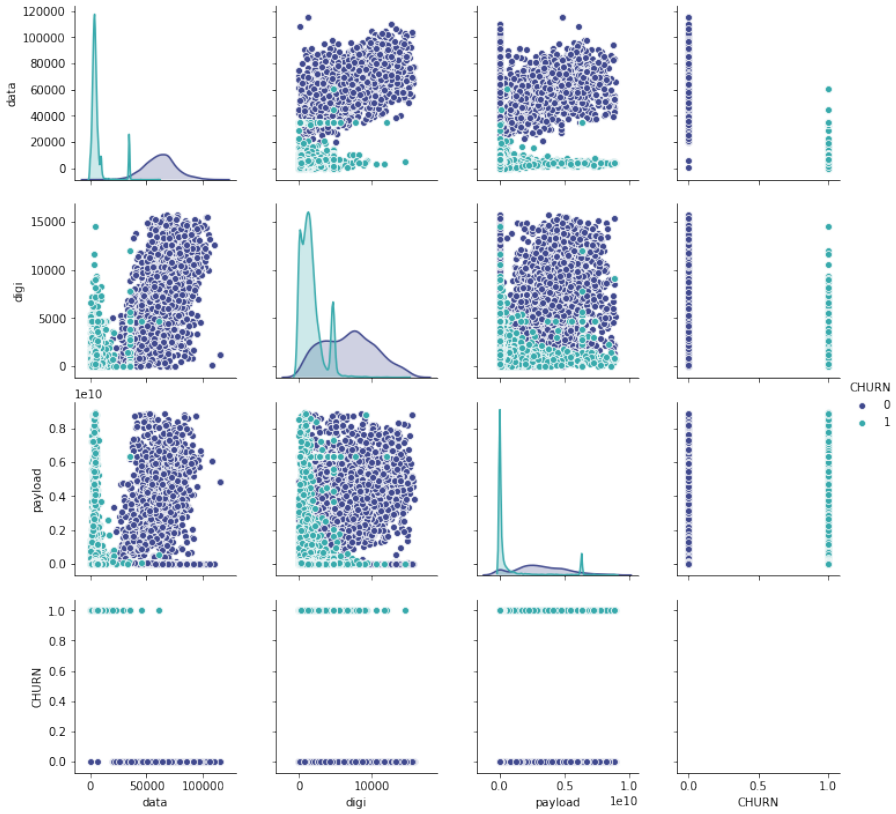
Pada bagian ini akan dijelaskan mengenai visualisasi data *input* dan hubungannya terhadap data status pelanggan. Dapat dilihat pada Gambar 4.3 terkait hubungan antara data *input* variabel *cluster, voice*, dan *sms* dengan data *output* variabel *churn*. Hubungan antara *cluster* dengan *cluster*, diperoleh informasi bahwa *churn* dan *non-churn* memiliki konsentrasi paling tinggi pada *cluster* wilayah nomor 3 (area Bogor), kemudian disusul dengan *cluster* wilayah nomor 1 (area Cibubur), dan terakhir *cluster* wilayah nomor 2 (area Depok). Selanjutnya, hubungan antara variabel *voice* dengan *voice*, diperoleh informasi bahwa pelanggan *churn* memiliki kecenderungan pengeluaran untuk jasa *voice* memiliki pengeluran di rentang 0 hingga 25000 rupiah dan terkonsentrasi di angka 10000 rupiah, sedangkan pelanggan *non-churn* memiliki kecenderungan pengeluaran untuk jasa *voice* di rentang 0 hingga 60000 rupiah dan terkonsentrasi di angka 30000 rupiah. Selanjutnya, hubungan antara variabel sms dengan sms, diperoleh informasi bahwa pelanggan *churn* memiliki kecenderungan pengeluaran untuk jasa *voice* memiliki pengeluran di rentang 0 hingga 10000 rupiah dan terkonsentrasi di angka sekitar 5000 rupiah, sedangkan pelanggan *non-churn* memiliki kecenderungan pengeluaran untuk jasa *voice* di rentang 0 hingga 20000 rupiah dan terkonsentrasi di angka sekitar 8000 dan sekitar 15000 rupiah. Terakhir, hubungan antara variabel *sms* dan variabel *voice*, pelanggan *churn* memiliki karakteristik cenderung terkonsentrasi di bagian kiri bawah diagram, sedangkan pelanggan *non-churn* memiliki karakteristik tersebar.



Gambar 4. 4 Hubungan antara Data *Input* Variabel *Cluster*, *Voice*, dan SMS serta Data *Output* Variabel *Churn*

Selanjutnya dapat dilihat pada Gambar 4.4 terkait hubungan antara data *input* variabel *data, digi*, dan *payoload* dengan data *output* variabel *churn*. Hubungan antara variabel *data* dengan *data*, diperoleh informasi bahwa karakteristik pengeluaran pelanggan *churn* terkonsentrasi pada angka sekitar 65000 rupiah, kemudian karakteristik pelanggan *non-churn* terkonsentrasi sebagian besar pada angka sekitar 5000 rupiah dan sebagian kecil sisanya terkonsentrasi pada angka 30000 rupiah. Selanjutnya, hubungan antara variabel *digi* dengan *digi*, diperoleh informasi bahwa karakteristik pengeluaran pelanggan *churn* terkonsentrasi sebagian besar pada angka sekitar 4000 hingga 5000 rupiah dan sebagian kecil sisanya terkonsentrasi pada angka 6000 rupiah, sedangkan karakteristik pelanggan *non-churn* terkonsentrasi pada angka 8000 rupiah. Terakhir, hubungan antara variabel *payload* dengan *payload,* diperoleh informasi

bahwa karakteristik penggunaan kuota pelanggan *churn* terkonsentrasi pada angka sekitar 5e+09 *byte* atau sekitar 0.5 *gigabyte* per bulan sedangkan karakteristik penggunaan kuota pelanggan *non-churn* cenderung tersebar mulai dari 0 hingga 10 *gigabyte* perbulan.



Gambar 4. 5 Hubungan antara Data *Input* Variabel *Data*, *Digital*, dan *Payload* serta Data *Output* Variabel *Churn*

(Halaman ini sengaja dikosongkan)

BAB 5
PEMBUATAN DAN ANALISA MODEL

Pada bab ini akan dilakukan analisa terkait variabel yang dinilai paling signifikan dalam mempengaruhi probabilitas seorang pelanggan *churn* atau *non-churn*. Selain itu, juga dilakukan pembuatan model sekaligus analisa terhadap hasil model yang dibangun dengan *tuning* parameter pada masing-masing metode.

5.1 Analisa Signifikansi Variabel *Input*

Pada bagian ini akan dijelaskan mengenai signifikansi masing-masing variabel terhadap variabel respon. Dari Gambar 4.2 diperoleh informasi mengenai *p-value* dan koefisien masing-masing variabel dengan menggunakan pendekatan salah satu metode dalam regresi logistik yaitu *logit* sebelum dilakukan normalisasi. Perhitungan dilakukan sebelum dilakukan normalisasi agar menghasilkan model prediksi yang lebih *robust* dan tidak terpengaruh dengan rentang maksimal dan atau minimal yang terdapat dalam normalisasi.

Selanjutnya, melihat *p-value* masing-masing variabel dengan tingkat *confidence level* di angka 0.95 diperoleh kesimpulan bahwa hanya terdapat variabel sebuah *independent variable* yang tidak cukup berpengaruh terhadap variabel respon yaitu variabel *sms* dengan *p-value* sebesar 0.94. Hal ini mengindikasikan bahwa variabel ini tidak perlu dilibatkan didalam pembuatan model.

Selanjutnya, interpretasi terhadap koefisien dari variabel *cluster* diperoleh kesimpulan bahwa pada pelanggan dengan masing-masing wilayah cibubur (label 1), depok (label 2), dan bogor (label 3) memiliki pengaruh positif terhadap peluang *churn* sebesar 3.76, 7.52, dan 11.28. Kemudian, pada variabel *voice* diperoleh kesimpulan bahwa peningkatan pengeluaran pelanggan 1 rupiah menurunkan peluang pelanggan *churn* sebesar 1×10^{-5}. Kemudian, interpretasi terhadap koefisien variabel *data* diperoleh kesimpulan peningkatan pengeluaran pelanggan 1 rupiah menurunkan peluang pelanggan *churn* sebesar 4×10^{-5}. Selanjutnya, hasil interpretasi terhadap koefisien variabel *digital* atau *digi* diperoleh kesimpulan bahwa peningkatan pengeluaran pelanggan 1 rupiah meningkatkan peluang pelanggan *churn* sebesar 4×10^{-5}. Selanjutnya, hasil interpretasi terhadap koefisien

variabel *payload* diperoleh kesimpulan peningkatan penggunaan pelanggan 1 *byte* menurunkan peluang pelanggan *churn* sebesar < 1 x 10^{-5}.

Kemudian, dari kelima variabel berikut dilakukan normalisasi terhadap data untuk mengetahui variabel yang paling signifikan pengaruhnya terhadap variabel respon. Berdasarkan Gambar 5.1 Diperoleh kesimpulan bahwa variabel yang paling signifikan memberikan perubahan terhadap kemungkinan pelanggan *churn* adalah variabel *data* mengingat variabel ini memiliki nilai koefisien mutlak setelah normalisasi yang paling besar dengan nilai koefisien sebesar -36,0498. Selain itu, diperoleh kesimpulan bahwa variabel data merupakan satu-satunya variabel yang memiliki hubungan negatif terhadap terjadinya pelanggan *churn* untuk brand Y. Sedangkan variabel *voice, sms, digital,* dan *payload* memiliki korelasi positif terhadap terjadinya pelanggan *churn*.

Results: Logit

Model:	Logit	Pseudo R-squared:	0.781
Dependent Variable:	CHURN	AIC:	2068.2580
Date:	2019-01-06 12:37	BIC:	2102.3725
No. Observations:	6788	Log-Likelihood:	-1029.1
Df Model:	4	LL-Null:	-4704.7
Df Residuals:	6783	LLR p-value:	0.0000
Converged:	1.0000	Scale:	1.0000
No. Iterations:	10.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
x1	4.3497	0.1596	27.2563	0.0000	4.0369	4.6625
x2	5.5037	0.5015	10.9738	0.0000	4.5207	6.4867
x3	-36.0498	1.3444	-26.8152	0.0000	-38.6847	-33.4148
x4	5.7716	0.4909	11.7572	0.0000	4.8095	6.7338
x5	8.7807	0.4692	18.7129	0.0000	7.8610	9.7003

Gambar 5. 1 Hasil Perhitungan dengan *Logit Model* terhadap *Independent Variable* dan *Dependent Variable* setelah Dilakukan Normalisasi

5.2 Pembuatan dan Analisa Model Prediksi

Regresi Logistik, *Support Vector Machine*, dan *Naive Bayes* merupakan beberapa metode yang berguna dalam mengetahui pola data *input* untuk selanjutnya dibangun sebuah model prediksi. Ketiga metode tersebut juga tergolong kedalam *supervised learning* dimana dibutuhkan *training* terhadap data *input*. Pembuatan model dilakukan pada aplikasi Anaconda Navigator, *environment* Spyder, dengan bahasa pemrograman *Python. Output* yang dihasilkan dari software ini berupa nilai

y hasil prediksi beserta performansi model *classifier* yang terdiri dari: tingkat akurasi, *recall, precision, f-1 score*, dan MR (*Missclassification Rate*).

Pada pendekatan *support vector machine,* terdapat tiga macam kernel yang digunakan dimana untuk masing-masing kernel akan dilakukan *tuning parameter* untuk mengetahui dampak dari *kernel* serta *parameter* yang disetel. Kernel yang diamati pada penelitian kali ini yaitu: kernel linear, RBF, dan polinomial. Parameter yang disetel pada metode *support vector machine* dengan kernel linear, sebatas pada nilai C, parameter yang disetel pada metode *support vector machine* dengan kernel RBF, terdiri dari nilai *slack* dan gamma, dan parameter yang disetel pada metode *support vector machine* dengan kernel polinomial terdiri dari nilai *slack* dan nilai derajat (d). Perhitungan nilai performansi dari setiap model dikombinasikan dengan 10-*fold cross validation* untuk memvalidasi nilai rata-rata performansi dari masing-masing model. Tabel 5.1 memberikan informasi mengenai macam *tuning* parameter yang dilakukan pada penelitian tugas akhir ini.

Tabel 5. 1 Macam-Macam Parameter yang Disetel Untuk Membangun Model *Classifier*

Metode	Kernel	C	Parameter	Model
Regresi Logistik	-	1	L1	U1
			L2	U2
Support Vector Machine	Linear	0.1	-	U3
		1	-	U4
		10	-	U5
		100	-	U6
	RBF	0.1	0.1	U7
		0.1	0.25	U8
		0.1	0.5	U9
		0.1	0.75	U10
		1	0.1	U11
		1	0.25	U12
		1	0.5	U13
		1	0.75	U14
		10	0.1	U15
		10	0.25	U16
		10	0.5	U17
		10	0.75	U18
		100	0.1	U19
		100	0.25	U20
		100	0.5	U21
		100	0.75	U22

Metode	Kernel	C	Parameter	Model
	Polinomial	0.1	2	U23
		0.1	3	U24
		0.1	5	U25
		0.1	7	U26
		1	2	U27
		1	3	U28
		1	5	U29
		1	7	U30
		10	2	U31
		10	3	U32
		10	5	U33
		10	7	U34
		100	2	U35
		100	3	U36
		100	5	U37
		100	7	U38
Naive Bayes	-	-	-	U39

Selanjutnya pada bagian ini sekaligus akan dilakukan *10-fold cross validation* untuk masing-masing metode guna memvalidasi performansi dari masing-masing model. Setiap metode dilakukan prediksi pelanggan *churn* sebanyak 10 kali dengan data *training* dan data *testing* yang berbeda-beda namun tetap bagian dari data *training* awal (setelah melalui tahap *data preprocessing*). Kemudian dilakukan rata-rata terhadap hasil dari 10 prediksi tadi untuk tiap-tiap parameter performansi. Terdapat 5 parameter performansi (*accuracy, recall, precision, f1-score*, dan *mean-squared-error*) yang akan diamati dalam penelitian tugas akhir ini, namun hanya akan terdapat 3 parameter yang digunakan dalam menentukan performansi sebuah model dalam penelitian tugas akhir ini yaitu: *accuracy, f1-score*, dan *missclassification rate* untuk masing-masing metode.

5.2.1 Pembuatan dan Analisa Model Classifier dengan Pendekatan Regresi Logistik

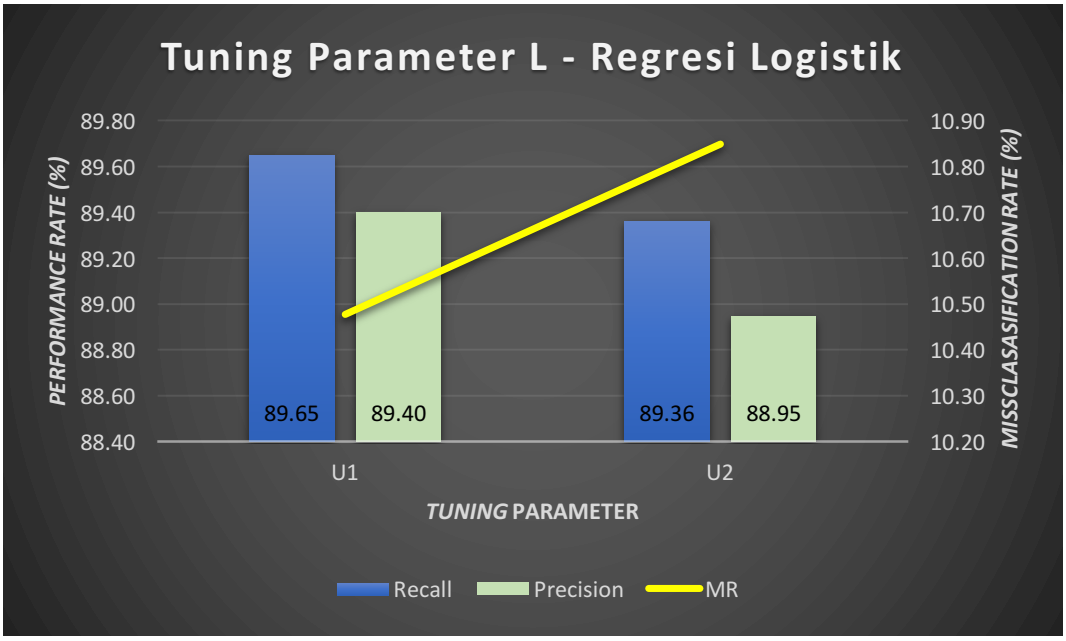
Terdapat satu macam *tuning* parameter yang digunakan pada pendekatan metode regresi logistik sebagai *input* kedalam *software* Ananconda Navigator dengan dua jenis model yang dibangun. Regresi logistik diketahui memiliki 2 macam regularisasi yaitu: L1, *manhattan distance,* atau juga diketahui sebagai LAE (*Least Absolute Errors*) dan L2, *euclidean distance,* atau juga diketahui sebagai

LSD (*Least Squared Errors*) dengan nilai C (*slack*) senilai 1. Parameter regularisasi sendiri pada regresi logistik berperan dalam mengukur jarak antar satu data dengan yang lainnya. Pada Tabel 5.2 diperoleh rata-rata performansi yang diperoleh dari pembuatan model *classifier* dengan pendekatan regresi logistik.

Tabel 5. 2 Nilai Performansi Model dengan Pendekatan Metode Regresi Logistik

Parameter	Rata-Rata Hasil (%)					Model
	Accuracy	Recall	Precision	F1-Score	MR	
L1	89.52	89.65	89.40	89.52	10.48	U1
L2	89.15	89.36	88.95	89.15	10.85	U2

Selanjutnya dilihat pada Gambar 5.2 tekait perbandingan tingkat *recall* dan *precision* pada model U1 dan U2. Diperoleh bahwa model U1 dengan regularisasi L1 mampu melakukan prediksi terhadap pelanggan *churn* lebih baik dari keseluruhan pelanggan yang diprediksi dibandingkan dengan menggunakan regularisasi L2. Selain itu model *classifier* dengan regularisasi L1 mampu memberikan tingkat kesalahan yang lebih rendah dibandingkan dengan model *classifier* yang menggunakan regularisasi L2. Secara keseluruhan *model classifier* U1 (regularisasi L1) memberikan hasil yang lebih baik pada pendekatan metode regresi logistik, serta mampu memprediksi pelanggan yang berpotensi untuk menghentikan layanannya di PT X dengan tingkat *recall* sebesar mencapai 89.65%. Hal ini dapat disebabkan salah satunya penggunaan parameter L1 lebih sesuai dengan karakteristik data yang sedang diolah.



Gambar 5. 2 Grafik Rata-Rata Performansi dan *Missclassification Rate* Model dengan Pendekatan Regresi Logistik

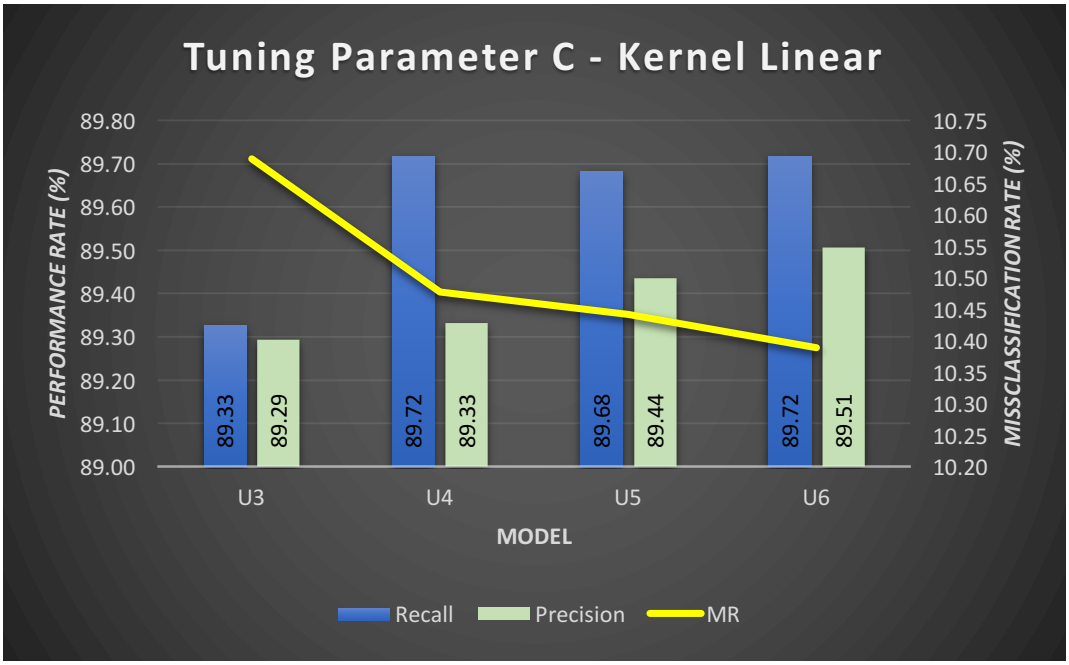
5.2.2 Pembuatan dan Analisa Model Classifier dengan Pendekatan Support Vector Machine, Kernel Linear

Pada pendekatan metode *support vector machine* untuk kernel linear terdapat satu macam parameter yang disetel dan diamati sebagai *input* kedalam *software*. Parameter yang disetel merupakan nilai C (*slack*) dengan nilai diantaranya: 0.1, 1, 10, dan 100. Masing-masing model diterapkan *10-cross validation* untuk memvalidasi nilai performansi *model classifier* yang dibangun. Pada Tabel 5.3 diperoleh rata-rata nilai performansi yang diperoleh dari pembuatan model *classifier* dengan pendekatan SVM, kernel linear.

Tabel 5. 3 Nilai Performansi Model dengan Pendekatan Metode SVM, Kernel Linear

Kernel	C	Rata-Rata Hasil (%)					Model
		Accuracy	Recall	Precision	F1-Score	MR	
Linear	0.1	89.31	89.33	89.29	89.31	10.69	U3
	1	89.52	89.72	89.33	89.52	10.48	U4
	10	89.56	89.68	89.44	89.56	10.44	U5
	100	89.61	89.72	89.51	89.61	10.39	U6

Dapat dilihat pada Gambar 5.3, peningkatan nilai *slack* secara eksponensial mampu meningkatkan rata-rata performansi *recall* dan *precision* yang dihasilkan. Kemudian dapat dilihat pada Gambar 5.3, peningkatan nilai variabel *slack* dapat menurunkan rata-rata tingkat *missclasification rate* dari model yang dihasilkan. Hal ini mengidikasikan bahwa nilai variabel *slack* merupakan *trade off* antara *error* dan *penalty* dimana ketika nilai variabel *slack* yang rendah cenderung menghasilkan *error* yang lebih tinggi sedangkan variabel *slack* yang tinggi cenderung menghasilkan *error* yang lebih rendah. Model yang dibangun dnengan dengan nilai MR yang lebih tinggi sebelum dilakukan penambahan nilai variabel *slack* mengindikasikan model tersebut masuk dalam kategori *underfitting*. Selanjutnya ketika penambahan nilai variabel *slack* memberikan tingkat *error* yang lebih tinggi dibandingkan variabel *slack* sebelumnya, hal tersebut mengindikasikan bahwa model telah mencapai *overfitting*. Selain itu, diperoleh kesimpulan bahwa model *classifier* U6 (penggunaan parameter C = 100) memberikan tingkat *error* paling rendah serta tingkat *recall* paling tinggi dibandingkan model lainnya pada kernel yang sama dengan tingkat *recall* dan *precision* masing-masing mencapai 89.72% dan 89.51%.



Gambar 5. 3 Grafik Rata-Rata Performansi dan *Missclassification Rate* Model dengan Pendekatan *Support Vector Machine*, kernel Linear

5.2.3 Pembuatan dan Analisa Model dengan Pendekatan Support Vector Machine, Kernel RBF

Pada pendekatan metode *support vector machine* dengan kernal RBF, teradapat dua jenis paramater yang disetel perubahannya yaitu nilai variabel *slack* dan nilai gamma. Parameter nilai *slack* (C) yang disetel memiliki nilai diantaranya: 0.1, 1, 10, dan 100 serta nilai gamma yang disetel memiliki nilai diantaranya: 0.1, 0.25, 0.5, dan 0.75. Pada Tabel 5.4 diperoleh rata-rata nilai performansi yang diperoleh dari pembuatan model *classifier* dengan pendekatan SVM, kernel RBF.

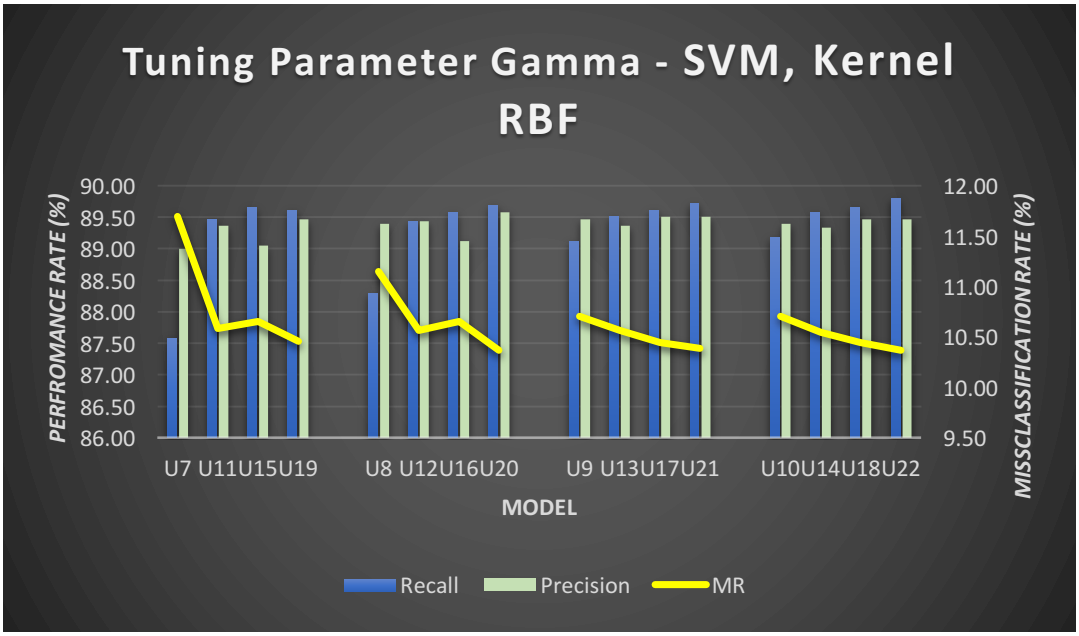
Tabel 5. 4 Nilai Performansi Model dengan Pendekatan Metode SVM, Kernel RBF

Kernel	C	Gamma	Rata-Rata Hasil (%)					Model
			Accuracy	Recall	Precision	F1-Score	MR	
RBF	0.1	0.10	88.30	87.59	89.00	88.29	11.70	U7
	0.1	0.25	88.85	88.30	89.39	88.84	11.15	U8
	0.1	0.50	89.29	89.11	89.47	89.29	10.71	U9
	0.1	0.75	89.29	89.19	89.40	89.29	10.71	U10
	1	0.10	89.42	89.47	89.37	89.42	10.58	U11
	1	0.25	89.43	89.43	89.43	89.43	10.57	U12
	1	0.50	89.43	89.50	89.36	89.43	10.57	U13
	1	0.75	89.45	89.58	89.33	89.45	10.55	U14
	10	0.10	89.35	89.65	89.05	89.35	10.65	U15
	10	0.25	89.35	89.58	89.12	89.35	10.65	U16

Kernel	C	Gamma	Rata-Rata Hasil (%)					Model
			Accuracy	Recall	Precision	F1-Score	MR	
	10	0.50	89.56	89.61	89.51	89.56	10.44	U17
	10	0.75	89.56	89.65	89.47	89.56	10.44	U18
	100	0.10	89.54	89.61	89.47	89.54	10.46	U19
	100	0.25	89.63	89.68	89.58	89.63	10.37	U20
	100	0.50	89.61	89.72	89.51	89.61	10.39	U21
	100	0.75	89.63	89.79	89.47	89.63	10.37	U22

Dapat dilihat pada Gambar 5.4, model dikelompokkan berdasarkan nilai gamma agar dapat diamati pengaruh dari nilai variabel *slack* terhadap performansi model yang dihasilkan dari pendekatan metode *Support Vector Machine*, kernel RBF. Parameter yang disetel pada model U7, U11, U15, dan U19 memiliki nilai gamma sebesar 0.1 dan nilai variabel *slack* masing-masing sebesar 0.1, 1, 10, dan 100. Kemudian, parameter yang disetel pada model U8, U12, U16, dan U20 memiliki nilai gamma sebesar 0.25 dan nilai variabel *slack* masing-masing sebesar 0.1, 1, 10, dan 100. Selanjutnya, parameter yang disetel pada model U9, U13, U17, dan U21 memiliki nilai gamma sebesar 0.5 dan nilai variabel *slack* masing-masing sebesar 0.1, 1, 10, dan 100. Terakhir, Parameter yang disetel pada model U10, U14, U18, dan U22 memiliki nilai gamma sebesar 0.75 dan nilai variabel *slack* masing-masing sebesar 0.1, 1, 10, dan 100.

Dari Gambar 5.4 diperoleh kesimpulan bahwa peningkatan nilai variabel *slack* dapat menurunkan *misclassification rate* yang dihasilkan. Selain itu, peningkatan nilai variabel *slack* dapat meningkatkan kemampuan model dalam melakukan prediksi pelanggan *churn* dengan tepat (*recall*). Namun, penambahan nilai variabel *slack* tidak sepenuhnya meningkatkan rata-rata *precision* dari model yang dihasilkan, pada U15, U12, U13, dan U14 menghasilkan rata-rata *precision* yang lebih rendah dibandingkan dengan nilai variabel *slack* pada model U11, U12, U9, dan U10 dengan nilai variabel *slack* yang lebih kecil.

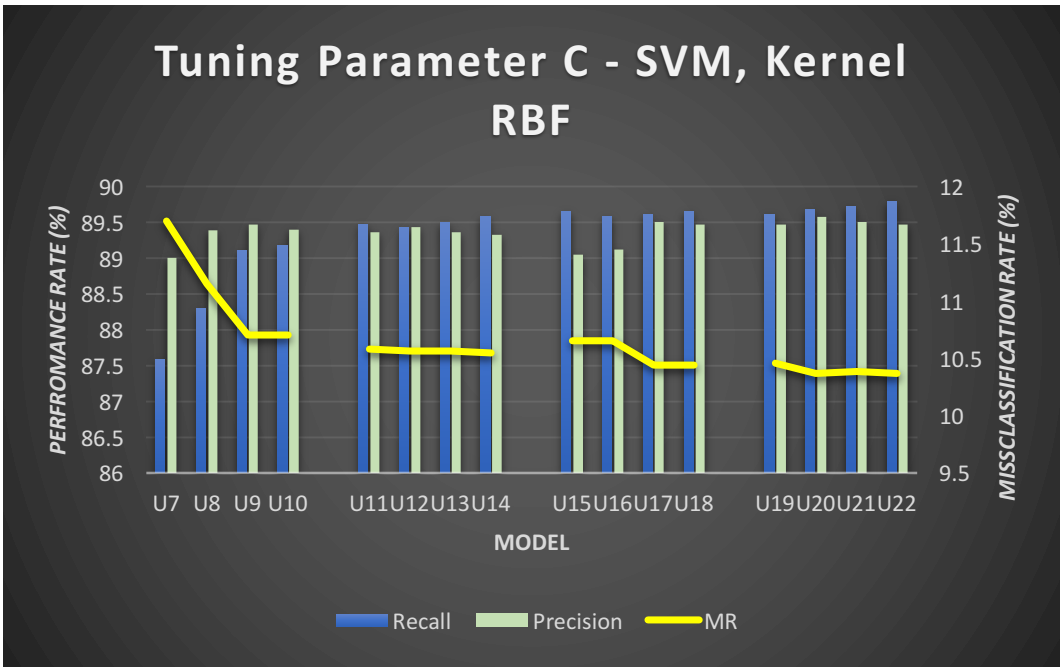


Gambar 5. 4 Grafik Rata-Rata Performansi dan *Missclassification Rate* yang Dikelompokkan Berdasarkan Nilai Variabel *Slack* dengan Pendekatan *Support Vector Machine*, kernel RBF

Selanjutnya, dapat dilihat pada Gambar 5.5, model dikelompokkan berdasarkan nilai variabel *slack* agar dapat diamati pengaruh dari perubahan nilai gamma terhadap performansi model yang dihasilkan dari pendekatan metode *Support Vector Machine*, kernel RBF. Parameter yang disetel pada model U7, U8, U9, dan U10 memiliki nilai variabel *slack* sebesar 0.1 dan nilai gamma masing-masing sebesar 0.1, 0.25, 0.5, dan 0.75. Kemudian, parameter yang disetel pada model U11, U12, U13, dan U14 memiliki nilai variabel *slack* sebesar 1 dan nilai gamma masing-masing sebesar 0.1, 0.25, 0.5, dan 0.75. Selanjutnya, parameter yang disetel pada model U15, U16, U17, dan U18 memiliki nilai variabel *slack* sebesar 10 dan nilai gamma masing-masing sebesar 0.1, 0.25, 0.5, dan 0.75. Terakhir, parameter yang disetel pada model U19, U20, U21, dan U22 memiliki nilai variabel *slack* sebesar 100 dan nilai gamma masing-masing sebesar 0.1, 0.25, 0.5, dan 0.75.

Dari gambar 5.5 juga diperoleh kesimpulan bahwa peningkatan nilai variabel gamma dapat menurunkan *missclassification rate* yang dihasilkan di semua nilai *slack*. Kemudian, penambahan nilai gamma pada nilai C = 0.1 dapat meningkatkan rata-rata *recall* secara signifikan. Namun, penambahan nilai gamma pada nilai C = 1, 10, dan 100, rata-rata *recall* yang cenderung stagnan atau tidak

berubah. Gamma mengindikasikan jumlah iterasi dalam pembuatan model dimana semakin semakin tinggi nilai *gamma,* iterasi yang dilakukan semakin tinggi sehingga rata-rata tingkat akurasi yang dilakukan lebih baik atau rata-rata tingkat MR yang dihasilkan lebih rendah. Sehingga, peningkatan nilai gamma dapat meningkatkan kemampuan model dalam melakukan prediksi pelanggan *churn* dengan tepat (*recall*). Namun, pada nilai gamma sebesar 0.75, tingkat *recall* yang dihasilkan cenderung lebih rendah dibandingkan dengan nilai gamma 0.5 di semua nilai *slack.* Hal ini dapat mengindikasikan terjadinya *overfitting* pada model dengan nilai gamma di angka tersebut. Terakhir, dari keseluruhan model yang dibangun dengan pendekatan SVM, kernel RBF, diperoleh model U22 sebagai model yang paling optimal dalam memprediksi pelanggan yang berpotensi *churn* atau berhenti menggunakan layanan dari PT X dengan tingkat kemampuan prediksi model sebesar 89.79% dan rata-rata tingkat *error* sebesar 10.37%.



Gambar 5. 5 Grafik Rata-Rata Performansi dan Missclassification Rate yang Dikelompokkan Berdasarkan Nilai Gamma dengan Pendekatan Support Vector Machine, kernel RBF

5.2.4 Pembuatan dan Analisa Model dengan Pendekatan Support Vector Machine, Kernel Polinomial

Pada pendekatan metode *support vector machine* untuk kernel polinomial terdapat dua macam parameter yang disetel dan diamati sebagai *input* kedalam

software. Parameter yang disetel merupakan nilai C (*slack*) dengan nilai diantaranya: 0.1, 1, 10, dan 100 dan nilai *d* (derajat) dengan nilai diantaranya: 2, 3, 5, dan 7. Masing-masing model diterapkan *10-cross validation* untuk memvalidasi nilai performansi *model classifier* yang dibangun. Pada Tabel 5.5 diperoleh rata-rata performansi yang diperoleh dari pembuatan model *classifier* dengan pendekatan SVM, kernel polinomial.

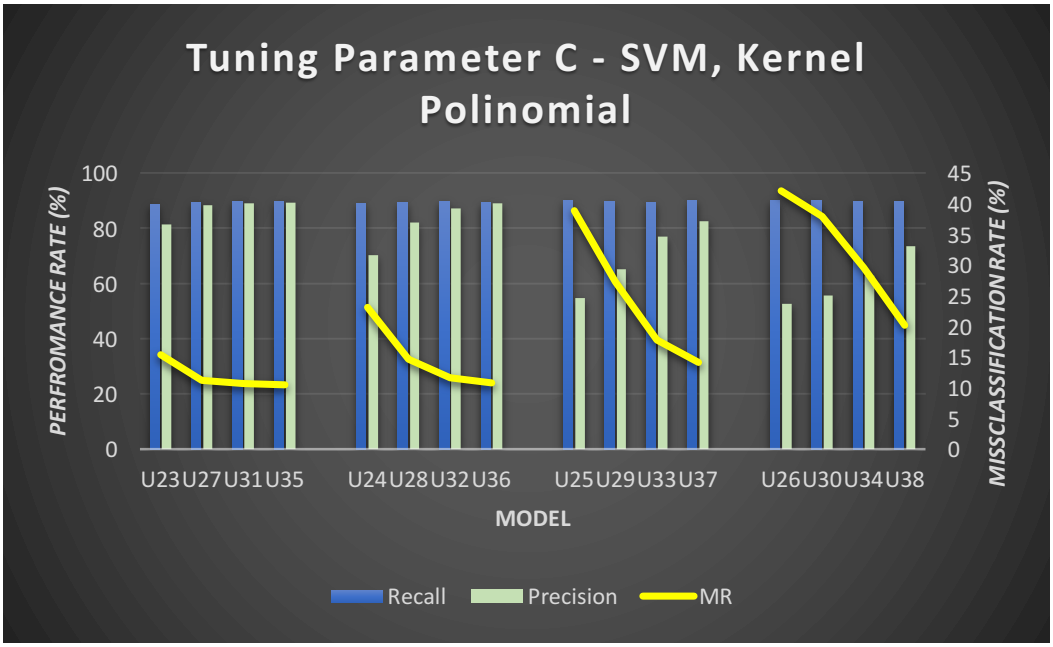
Tabel 5. 5 Nilai Performansi Model dengan Pendekatan Metode SVM, Kernel Polinomial

Kernel	C	d	Rata-Rata Hasil (%)					Model
			Accuracy	Recall	Precision	F1-Score	MSE	
Polinomial	0.1	2	84.63	88.55	81.45	84.85	15.37	U23
	0.1	3	76.94	88.94	70.22	78.47	23.06	U24
	0.1	5	61.11	90.00	54.79	68.11	38.89	U25
	0.1	7	57.90	90.00	52.53	66.33	42.10	U26
	1	2	88.83	89.36	88.32	88.84	11.17	U27
	1	3	85.40	89.54	82.02	85.61	14.60	U28
	1	5	72.71	89.68	65.11	75.44	27.29	U29
	1	7	62.10	89.96	55.54	68.67	37.90	U30
	10	2	89.33	89.65	89.02	89.33	10.67	U31
	10	3	88.39	89.68	87.19	88.41	11.61	U32
	10	5	82.20	89.61	76.95	82.79	17.80	U33
	10	7	70.47	89.72	62.81	73.89	29.53	U34
	100	2	89.51	89.72	89.30	89.51	10.49	U35
	100	3	89.26	89.57	88.95	89.26	10.74	U36
	100	5	85.88	89.86	82.58	86.06	14.12	U37
	100	7	79.82	89.68	73.58	80.82	20.18	U38

Dapat dilihat pada Gambar 5.6, model dikelompokkan berdasarkan nilai parameter *d* agar dapat diamati pengaruh dari nilai variabel *slack* terhadap performansi model yang dihasilkan dari pendekatan metode *Support Vector Machine*, kernel Polinomial. Parameter yang disetel pada model U23, U27, U31, dan U35 memiliki nilai d sebesar 2 dan nilai variabel *slack* masing-masing sebesar 0.1, 1, 10, dan 100. Kemudian, parameter yang disetel pada model U24, U28, U32, dan U36 memiliki nilai d sebesar 3 dan nilai variabel *slack* masing-masing sebesar 0.1, 1, 10, dan 100. Selanjutnya, parameter yang disetel pada model U25, U29, U33, dan U37 memiliki nilai d sebesar 5 dan nilai variabel *slack* masing-masing sebesar 0.1, 1, 10, dan 100. Terakhir, Parameter yang disetel pada model U26, U30, U34,

dan U38 memiliki nilai d sebesar 7 dan nilai variabel *slack* masing-masing sebesar 0.1, 1, 10, dan 100.

Dari Gambar 5.6, diperoleh kesimpulan bahwa peningkatan nilai variabel *slack* dapat menurunkan *misclassification rate* yang dihasilkan di semua nilai *d*. Selain itu peningkatan nilai variabel *slack* tidak berpengaruh signifikan terhadap nilai rata-rata *recall* yang dihasilkan, hal ini berlaku di semua nilai *d* dimana performansi semua model cenderung stagnan. Namun, penambahan nilai variabel *slack* dapat meningkatkan nilai rata-rata *precision* atau kemampuan model dalam memprediksi pelanggan *non-churn* dengan tepat terhadap total pelanggan *non-churn* yang diprediksi.



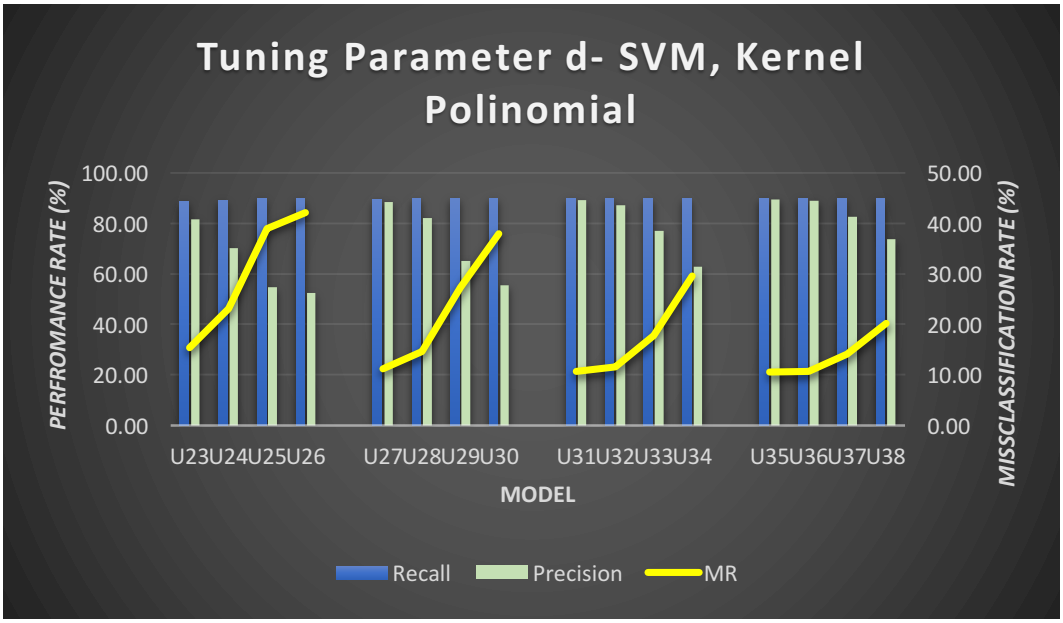
Gambar 5. 6 Grafik Rata-Rata Performansi dan *Missclassification Rate* yang Dikelompokkan Berdasarkan Nilai *d* dengan Pendekatan *Support Vector Machine*, kernel Polinomial

Selanjutnya, dapat dilihat pada Gambar 5.7, model dikelompokkan berdasarkan nilai variabel *slack* agar dapat diamati pengaruh dari perubahan nilai *d* terhadap performansi model yang dihasilkan dari pendekatan metode *Support Vector Machine*, kernel polinomial. Parameter yang disetel pada model U23, U24, U25, dan U26 memiliki nilai variabel *slack* sebesar 0.1 dan nilai *d* masing-masing sebesar 2, 3, 5, 7. Kemudian, parameter yang disetel pada model U27, U28, U29, dan U30 memiliki nilai variabel *slack* sebesar 1 dan nilai gamma masing-masing

sebesar 2, 3, 5, 7. Selanjutnya, parameter yang disetel pada model U31, U32, U33, dan U34, memiliki nilai variabel *slack* sebesar 10 dengan nilai *d* masing-masing sebesar 2, 3, 5, dan 7. Terakhir, parameter yang disetel pada model U35, U36, U37, U38 memiliki nilai variabel *slack* sebesar 100 dan nilai gamma masing-masing sebesar 2, 3, 5, dan 7.

Nilai *d* atau *degree* merupakan kemampuan *classifier* dalam mengikuti pola persebaran data. Semakin tinggi nilai *d* mengindikasikan kemampuan *classifier* menyerupai pola semakin besar. Namun, semakin tinggi kemampuan model dalam menyerupai pola data, dapat pula menyebabkan model yang *overfitting.* Dari Gambar 5.7 diperoleh kesimpulan bahwa peningkatan nilai variabel *d* dapat meningkatkan rata-rata nilai MR yang dihasilkan pada semua kelompok nilai variabel *slack*. Selain itu, peningkatan nilai *d* pada *slack* yang lebih rendah memberikan peningkatan *missclassification rate* yang lebih signifikan dibandingkan dengan peningkatan *slack* yang lebih tinggi. Hal ini dikarenakan, nilai *slack* yang lebih besar, memberikan toleransi yang lebih besar bagi terhadap data yang tidak berada dalam daerah kelas yang seharusnya sehingga *missclassification rate* yang dihasilkan juga lebih kecil.

Kemudian, penambahan nilai *d* menurunkan nilai *precision* dari model yang dibangun di semua nilai *slack.* Lebih jauh lagi, penurunan rata-rata nilai *precision* model pada nilai *slack* yang lebih rendah menyebabkan penurunan nilai *precision* lebih signifikan dibandingkan penurunan rata-rata nilai *precision* pada *slack* yang lebih tinggi. Namun, peningkatan nilai *d* tidak berpengaruh signifikan terhadap rata-rata nilai *recall* yang dihasilkan oleh model. Terakhir, dari pendekatan metode SVM, kernel polinomial diperoleh model U35 (model dengan nilai *slack* = 100 dan nilai *d* = 2) sebagai model yang paling optimal dalam memprediksi pelanggan yang berpotensi *churn* atau berhenti menggunakan layanan dari PT X dengan tingkat kemampuan prediksi model sebesar 89.72% dan rata-rata tingkat *error* sebesar 10.49%.



Gambar 5. 7 Grafik Rata-Rata Performansi dan *Missclassification Rate* yang Dikelompokkan Berdasarkan Nilai Slack dengan Pendekatan *Support Vector Machine*, kernel Polinomial

5.2.5 Pembuatan dan Analisa Model dengan Pendekatan Naïve Bayes

Dalam implementasi metode *naive bayes* hanya terdapat satu macam model. Terdapat beberapa kekurangan dalam metode ini yaitu data input harus bersifat diskrit. Selain itu, metode ini sensitif terhadap kejadian dengan probabilitas 0. Hal ini diperlukan *treatment* khusus berupa *clustering* untuk membangun data kategorikal pada data *input.* Pada aplikasi *Spyder* telah tersedia *library sklearn.naive_bayes.GaussianNaiveBayes* yang mengakomodasi data input yang bersifat kontinun untuk selanjutnya dirubah menjadi data kategorikal mengikuti distribusi normal atau Gaussian.

Dari hasil pengujian diperoleh rata-rata akurasi dengan metode *Naive Bayes* sebesar 74.34%, rata-rata *recall* sebesar 78.93%, rata-rata *precision* sebesar 74.55 %, rata-rata *f1-score* sebesar 75.95%, dan rata-rata *mean squared error* sebesar 25.66% Rendahnya kemampuan model dalam melakukan prediksi dapat disebabkan oleh variabel *input* yang memiliki korelasi antar variabel cukup tinggi sedangkan salah satu asumsi yang terdapat dalam metode ini tidak terdapatnya korelasi antar variabel *input*.

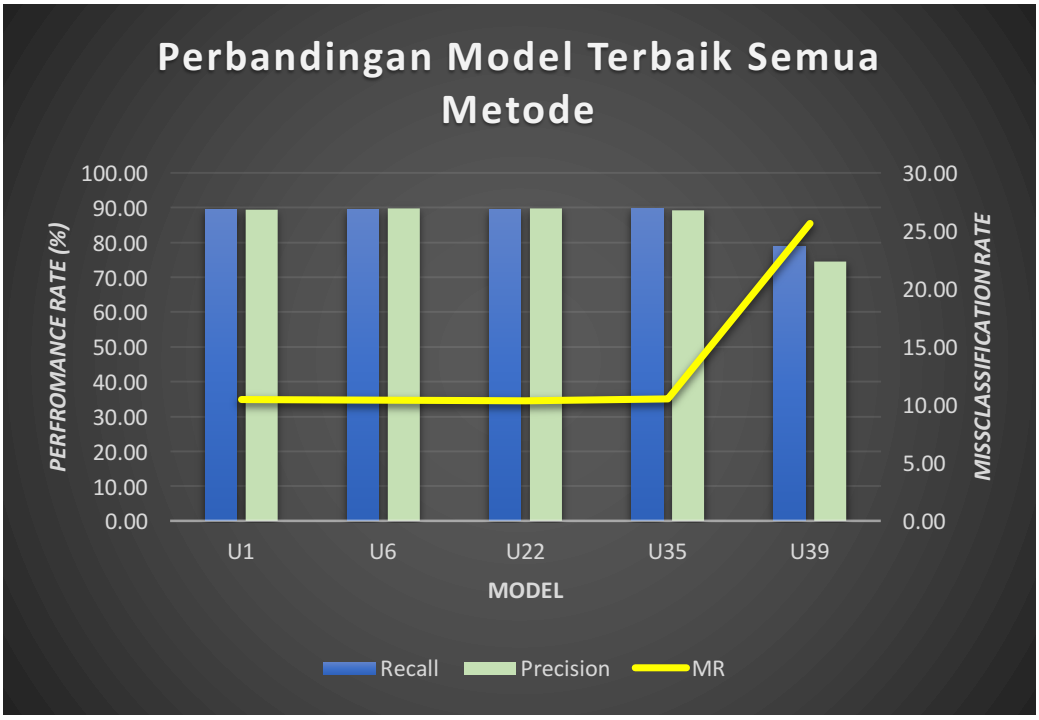
5.3 Perbandingan Model Terbaik dari Masing-Masing Metode

Pada bagian ini akan dibandingkan model terbaik dari masing-masing metode berdasarkan parameter yang telah dijabarkan sebelumnya. Pada metode Regresi Logistik, U1 (model dengan regularisasi L1) memiliki performansi yang lebih baik dibandingkan dengan U2 (model dengan regularisasi L2). Hal ini dibuktikan dengan tingkat *error* yang lebih rendah serta nilai *accuracy, recall,* dan *precision* yang lebih tinggi.

Selanjutnya pada metode *Support Vector Machine* dengan kernel Linear, U6 (model dengan nilai *slack* = 100) memiliki performansi terbaik dibandingkan model lainnya pada metode dan kernel yang sama. Hal ini dibuktikan dengan nilai *recall* dan *precision* yang lebih tinggi dibandingkan dengan skenario U3, U4, dan U5. Pada kernel RBF, model U22 (model dengan nilai *slack* = 100 dan nilai gamma = 0.75) memiliki nilai performansi *recall* dan *precision* yang paling baik, serta tingkat *error* yang paling rendah dibandingkan semua model pada metode dan kernel yang sama. Kemudian, dari kernel *Polinomial,* U35 memiliki nilai performansi *recall* yang baik serta *missclassification* yang paling rendah dibandingkan dengan model lainnya yang berada pada metode dan kernel yang sama. Hal tersebut dibuktikan dengan tingkat *recall* dan *precision* masing-masing mencapai 89.72% dan 89.30%. Meskipun nilai performansi *precision* masih dibawah U27, namun model U35 memberikan tingkat akurasi serta *recall* yang lebih baik. Selain itu, meskipun nilai *recall* U35 dibawah U25 dan U26, namun U25 dan U26 memiliki *misclassification rate* yang cukup tinggi dengan nilai masing-masing 54.79% dan 52.53%. Terakhir yaitu pada metode *Naive Bayes* diperoleh U39 sebagai model paling baik berdasarkan metode tersebut dengan tingkat *recall, precision,* dan *misclassification rate* masing-masing mencapai 78.93%, 74.55%, dan 25.66%.

Berdasarkan Gambar 5.8 dibawah diketahui model *classifier* terbaik dari masing-masing metode. Diketahui bahwa model terbaik yang dihasilkan dengan pendekatan Regresi Logistik dan *Support Vector Machine* memberikan tingkat rata-rata *recall, precision,* dan *misclassification rate* yang hampir serupa. Selain itu, diperoleh *recall, precision,* dan *misclassification rate* yang memiliki perbedaan cukup signifikan dibandingkan model lainnya. Dari gambar tersebut diperoleh

kesimpulan bahwa model *classifier* terbaik yang digunakan untuk prediksi pelanggan *churn* pada PT X merupakan model U22 dengan kernel RBF, nilai *slack* = 100, dan nilai gamma = 0.75 dengan tingkat rata-rata performansi *recall, precision,* serta *misclassification rate* masing-masing mencapai 89.72%, 89.30%, dan 10.49%.



Gambar 5. 8 Perbandingan Rata-Rata Performansi dan *Missclassification Rate* Antar Model Terbaik dari Masing-Masing Metode

5.4 Implementasi Metode dan Analisa Signifikansi terhadap Brand W dan Z pada PT X

Selanjutnya, setelah diperoleh metode optimal untuk memprediksi pelanggan berdasarkan *tuning parameter* pada bagian sebelumnya. dilakukan penerapan metode optimal tersebut untuk *brand* W dan Z dari PT X yang masing-masing sama-sama memiliki sifat pra-bayar.

5.4.1 Analisa Penerapan Metode dan Karakteristik Pelanggan pada Brand W

Hasil *running* model untuk *brand* W diperoleh rata-rata tingkat *recall, precision,* dan *misclassification rate* masing-masing sebesar 51.1%, 75.6%, dan 25.4%. Kemudian dilakukan signifikansi variabel terhadap probabilitas terjadinya pelanggan *churn* setelah dilakukan normalisasi. Berdasarkan Gambar 5.9 diperoleh

koefisien *voice* (x1), *sms*(x2), *data*(x3), digital(x4), dan *payload*(x5) masing-masing diantaranya: 16.78, -6.42, -26.85, 2.53, dan -3.79.

Dari informasi tersebut, dapat diperoleh informasi bahwa pelanggan yang termasuk dalam kategori *non-churn* memiliki karakteristik berupa *sms, data,* dan *payload* yang tinggi. Sedangkan, untuk pelanggan yang termasuk dalam kategori *churn* cenderung memiliki karakteristik penggunaan *voice* serta *digital* yang tinggi. Diketahui bahwa segmen pasar yang dituju dari *brand* ini yaitu kalangan pasar dengan rentang umur 12-19 tahun (Panji, 2014). Sehingga, dapat disimpulkan bahwa pelanggan yang loyal menggunakan layanan *brand* W dari PT X cenderung memiliki karakteristik sebatas menggunakan layanan data dan sebagian kecil *sms*.

Results: Logit

Model:	Logit	Pseudo R-squared:	0.262
Dependent Variable:	churn	AIC:	15640.5235
Date:	2019-01-10 00:14	BIC:	15679.1793
No. Observations:	16834	Log-Likelihood:	-7815.3
Df Model:	4	LL-Null:	-10585.
Df Residuals:	16829	LLR p-value:	0.0000
Converged:	1.0000	Scale:	1.0000
No. Iterations:	6.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
x1	16.7838	0.7450	22.5280	0.0000	15.3236	18.2440
x2	-6.4285	0.5164	-12.4497	0.0000	-7.4405	-5.4164
x3	-26.8469	0.6377	-42.0989	0.0000	-28.0968	-25.5971
x4	2.5314	0.8924	2.8365	0.0046	0.7823	4.2805
x5	-3.7936	0.7895	-4.8051	0.0000	-5.3409	-2.2462

Gambar 5. 9 Hasil Perhitungan dengan *Logit Model* terhadap *Independent Variable* dan *Dependent Variable* setelah Dilakukan Normalisasi pada *Brand* W

5.4.2 Analisa Penerapan Metode dan Karakteristik Pelanggan pada Brand Z

Hasil *running* model untuk *brand* Z diperoleh rata-rata tingkat *recall, precision,* dan *misclassification rate* masing-masing sebesar 51.1%, 75.6%, dan 25.4%. Kemudian dilakukan signifikansi variabel terhadap probabilitas terjadinya pelanggan *churn* setelah dilakukan normalisasi. Berdasarkan Gambar 5.9 diperoleh koefisien *voice* (x1), *sms*(x2), *data*(x3), digital(x4), dan *payload*(x5) masing-masing diantaranya: -5.23, 13.09, -13.29, -31.39, dan 277.76.

Dari informasi tersebut, dapat diperoleh informasi bahwa pelanggan yang termasuk dalam kategori *non-churn* memiliki karakteristik berupa *voice, data,* dan

digital yang tinggi. Sedangkan, untuk pelanggan yang termasuk dalam kategori *churn* cenderung memiliki karakteristik penggunaan *sms* serta *payload* yang tinggi. Diketahui bahwa segmen pasar yang dituju dari *brand Z* ini yaitu kalangan pasar di daerah *rural* atau desa (Burhani, 2009). Sehingga, dapat disimpulkan bahwa pelanggan yang loyal menggunakan layanan *brand* Z dari PT X cenderung memiliki karakteristik sebatas menggunakan layanan *voice, data,* dan *digital*.

Results: Logit

Model:	Logit	Pseudo R-squared:	0.247
Dependent Variable:	churn	AIC:	19445.1257
Date:	2019-01-10 08:46	BIC:	19484.4385
No. Observations:	19198	Log-Likelihood:	-9717.6
Df Model:	4	LL-Null:	-12913.
Df Residuals:	19193	LLR p-value:	0.0000
Converged:	1.0000	Scale:	1.0000
No. Iterations:	9.0000		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
x1	-5.2364	0.6405	-8.1754	0.0000	-6.4917	-3.9810
x2	13.0884	0.3918	33.4087	0.0000	12.3205	13.8562
x3	-13.2911	0.2589	-51.3396	0.0000	-13.7985	-12.7837
x4	-31.3967	3.2002	-9.8109	0.0000	-37.6689	-25.1244
x5	277.7645	14.2756	19.4573	0.0000	249.7850	305.7441

Gambar 5. 10 Hasil Perhitungan dengan *Logit Model* terhadap *Independent Variable* dan *Dependent Variable* setelah Dilakukan Normalisasi pada *Brand* W

(Halaman ini sengaja dikosongkan)

BAB 6
KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan mengenai kesimpulan serta saran yang diperoleh dari hasil pengerjaan tugas akhir ini. Pada bab ini juga akan dikemukakan beberapa saran yang berhubungan dengan pelaksanaan penelitian lebih lanjut.

6.1 Kesimpulan

Berikut merupakan kesimpulan terhadap pelaksanaan tugas akhir ini yang diantranya:

a) Perbandingan seluruh model diperoleh prediksi diperoleh menunjukkan bahwa pendekatan *Support Vector Machine*, dengan Kernel RBF, nilai *slack* 100, dan gamma 0.75 merupakan model terbaik dalam memprediksi pelanggan untuk brand Y
b) Variabel yang memiliki signifikansi paling tinggi dalam melakukan prediksi pelanggan *churn* merupakan variabel *data.* Selain itu, variabel *data* menjadi satu-satunya variabel yang memiliki korelasi negatif terhadap terjadinya pelanggan *churn* pada *brand* Y.
c) Setiap *brand* yang terdapat pada PT X memiliki karakteristik pasar masing-masing. Hal ini bertujuan salah satunya agar layanan dari PT X mampu memasuki semua segmen pasar yang tersedia.

6.2 Saran

Berikut merupakan saran terhadap penelitian selanjutnya:

a) Melibatkan lebih banyak variabel *independent* baru guna mencari performansi yang lebih baik. Hal tersebut dapat dikombinasikan dengan data frekuensi keluhan pelanggan, data demografi, serta karakteristik pendukung lainnya.
b) Menambahkan pendekatan metode serta *tuning* parameter yang lain guna memperkaya referensi model *classifier*.
c) Menambahkan jumlah data yang digunakan sebagai data *input* kedalam proses pembuatan model.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

Ahn, Jae-Hyeon., Sang-Pil Han, & Yung-Seop Lee. (2006). *Customer churn analysis Churn determinants and mediation effects of partial defection in the Korean mobile telecommunications service industry*. Department of Statistics, Dongguk University, Seoul, Korea. Elsevier Ltd.

Bligh, Philip, and Douglas Turk. (2004). *CRM Unplugged : Releasing CRM's Strategic Value*. Hoboken: John Wiley & Sons, Inc.

Chandra, Andreas. (2017). *Perbedaan Supervised and Unsupervised Learning*. [online] Tersedia pada: https://www.datascience.or.id/detail_artikel/52/supervised-and-unsupervised-learning . Diakses pada 22 September 2018.

Brinberg, Miriam. (2015). *Cross Validation Tutorial*. [*online*] Dapat diakses pada: https://quantdev.ssri.psu.edu/tutorials/cross-validation-tutorial. Diakses pada tanggal 1 Januari 2019

Burhani, Rusian. (2009). *Telkomsel Perluas Pasar Kartu AS.* [*online*] Dapat diakses pada: https://www.antaranews.com/berita/158926/telkomsel-perluas-pasar-kartu-as . Diakses pada tanggal 10 Januari, 2019.

Eremenko, Kirill & Hadelin de Ponteves. (2016). *Machine Learning A-Z^{TM}: Hands-On Python & R in Data Science.* [*online*] Dapat diakses pada : https://www.udemy.com/machinelearning/learn/v4/overview?siteID=eyzsD2QGsYg-hvNIlGtTuGhjvZmXGmc4zg&LSNPUBID=eyzsD2QGsYg . Diakses pada tanggal 24 September, 2018

Gronros, Lovisa. & Ida Janer. (2018). *Predicting Customer Churn Rate in the iGaming Industry using Supervised Machine Learning*. Stockholm, Swedia. KTH Royal Institute of Technology

Han, Jiawei et.al. (2011). *DATA MINING Concepts and Techniques*. 225 Wyman Street, Waltham, MA 02451, USA. Morgan Kaufmann Publishers

Kotler, P. (2003). *Marketing Management, 11th edition*. Prentice Hall, New Jersey.

Kim, M., Park, M., & Jeong, D. (2004). *The effects of customer satisfaction and switching barriers on customer loyalty in Korean Mobile Telecommunication Services*. Telecommunications Policy, 28(2), 145–159.

Panji, Aditya. (2014). *Telkom Rilis Kartu Perdana "Loop"*. [*online*] Dapat diakses pada: https://tekno.kompas.com/read/2014/03/10/1446318/Telkomsel.Rilis.Kartu.Perdana.Loop. Diakses pada 10 Januari 2019.

Park, Hyeoun. (2013). *An Introduction to Logistic Regression: From Basic Concepts to Interpretation with Particular Attention to Nursing Domain*. College of Nursing and System Biomedical Informatics Narional Core Research Center, Seoul National University, Korea. J Korean Acar Nurs

Peppers, D. & Rogers, M., (2004). *Managing Customer Relationships: A Strategic Framework*. Hoboken, New Jersey: John Wiley & Sons.

Rice, Scott R G. 2014. *The ARPU of Identity.* PacificEast Research

Rodpysh, Keyvan Vahidy, Amir Aghai, and Meysam Majdi. (2012). *APPLYING DATA MINING IN CUSTOMER RELATIONSHIP MANAGEMENT*. International Journal of Information Technologu, Control and Automation (IJITCA)

Santosa, Budi dan Ardian Umam. (2018). *Data Mining dan Big Data Analytics : Teori dan Implementasi Menggunakan Python & Apache Spark"*. Yogyakarta. Penebar Media Pustaka

Venkatesan, Rajkumar, Trichy V. Krishnan, and V. Kumar (2004), *"Evolutionary Estimation of Macro-Level Diffusion Models,"* Marketing Science, forthcoming

Wei, Chin-Ping and I-Tang Chiu. (2002). *Turning telecommunications call details to churn prediction: a data mining approach*. Taiwan. Pergamon

Yang, L.S. dan Chiu C. (2006). *"Knowledge Discovery on Customer Churn Prediction"*. Dallas, Texas, Amerika Serikat. Prosiding dalam 10th WSEAS International Confrence dibidang Matematika Terapan

Zaenuddin, Ahmad. (2017). *Bisnis Kartu Perdana yang Terusik oleh Aturan Registrasi SIM Card*. [online] : https://amp.tirto.id/bisnis-kartu-perdana-yang-terusik-oleh-aturan-registrasi-sim-card-czy4 . Diakses pada 24 Oktober 2018

LAMPIRAN

Lampiran A
Contoh Rekapan Data Sebelum Dilakukan *Data Preprocessing*

No	Region	Branch	Cluster	Kecamatan	Brand	Voice	SMS	Data	Digital	Payload	Status
1	EASTERN JABOTABEK	BRANCH KARAWANG	KARAWANG	TALAGASARI	W	4699	2029	2592	25	4699	G
2	EASTERN JABOTABEK	BRANCH BOGOR	BOGOR	JAMPANG TENGAH	Y	17745	9693	38466	6133	17745	A
3	EASTERN JABOTABEK	BRANCH KARAWANG	CIKARANG	SUKATANI	X	4699	3267	3338	754	4699	G
4	EASTERN JABOTABEK	BRANCH KARAWANG	PURWAKARTA	CAMPAKA	Z	19457	9718	49882	3398	19457	A
5	EASTERN JABOTABEK	BRANCH BOGOR	CIBUBUR	SUKAMAKMUR	X	22882	8762	32569	2326	22882	A
6	EASTERN JABOTABEK	BRANCH KARAWANG	PURWAKARTA	KIARAPEDES	Z	42647	13348	59859	13367	42647	A
7	EASTERN JABOTABEK	BRANCH BOGOR	BOGOR	TAJUR HALANG	Z	20668	5687	38357	1810	20668	A
8	EASTERN JABOTABEK	BRANCH BOGOR	BOGOR	CIAMPEA	Z	281	250	2049	25	281	G
9	EASTERN JABOTABEK	BRANCH BOGOR	BOGOR	JAMPANG KULON	W	190870	5832	0	150	190870	A
10	EASTERN JABOTABEK	BRANCH KARAWANG	KARAWANG	CILAMAYA KULON	X	6287	3000	5000	225	6287	G
11	EASTERN JABOTABEK	BRANCH BOGOR	DEPOK	PARUNG	X	5303	3109	4160	922	5303	G

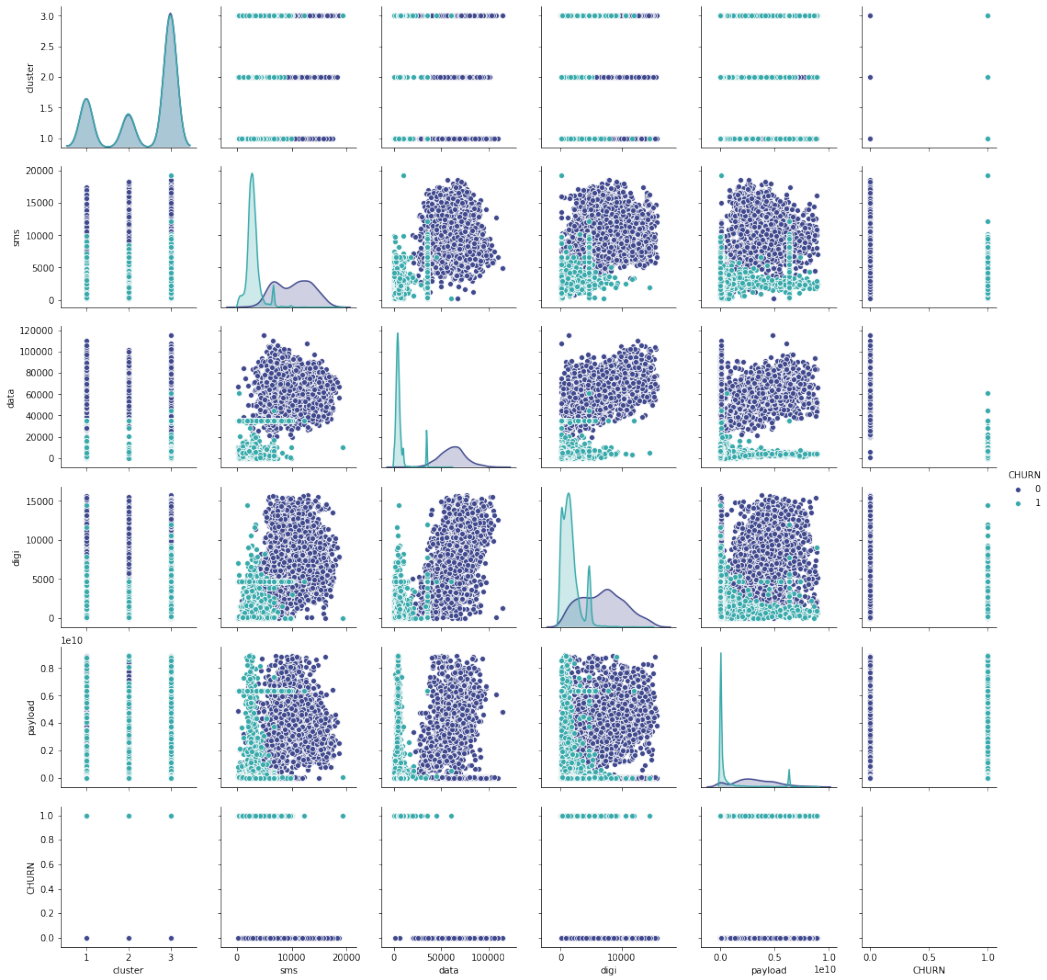
No	Region	Branch	Cluster	Kecamatan	Brand	Voice	SMS	Data	Digital	Payload	Status
12	EASTERN JABOTABEK	BRANCH BOGOR	BOGOR	PALABUHANRATU	W	80897	7457	18796	14176	80897	A
13	EASTERN JABOTABEK	BRANCH BOGOR	DEPOK	BOJONGSARI	X	38133	12578	64445	6815	38133	A
14	EASTERN JABOTABEK	BRANCH KARAWANG	PURWAKARTA	CAMPAKA	X	3988	4204	6555	763	3988	G
15	EASTERN JABOTABEK	BRANCH KARAWANG	CIKARANG	KARANGBAHAGIA	Z	27361	12546	29362	4440	27361	A
16	EASTERN JABOTABEK	BRANCH BOGOR	BOGOR	DRAMAGA	Y	0	169	101	25	0	G
17	EASTERN JABOTABEK	BRANCH BOGOR	BOGOR	LENGKONG	Y	20690	8285	30014	1247	20690	A
18	EASTERN JABOTABEK	BRANCH KARAWANG	KARAWANG	TALAGASARI	X	4699	2029	2592	25	4699	G
19	EASTERN JABOTABEK	BRANCH BOGOR	BOGOR	JAMPANG TENGAH	Y	17745	9693	38466	6133	17745	A
20	EASTERN JABOTABEK	BRANCH KARAWANG	CIKARANG	SUKATANI	X	4699	3267	3338	754	4699	G

Lampiran B
Contoh Rekapan Data setelah Dilakukan Normalisasi

Data	cluster	voice	sms	data	digital	payload
0	0	0.37153	0.44845	0.28281	0.14708	0.38151
1	0.5	0.08575	0.14975	0.03609	0.05753	0.00711
2	0	0.28106	0.51731	0.17426	0.18311	0.08529
3	1	0.28651	0.46626	0.40049	0.28203	0.65028
4	1	0.43547	0.51102	0.53876	0.40570	0.69023
5	1	0.43232	0.61765	0.56470	0.31303	0.85143
6	1	0.11773	0.17015	0.01846	0.10415	0.00249
7	1	0.32935	0.32238	0.53843	0.14516	0.16680
8	1	0.12131	0.31133	0.05346	0.00191	0.01180
9	1	0.68329	0.67910	0.63619	0.64819	0.36480
10	1	0.09042	0.18103	0.03708	0.04241	0.00130
11	1	0.39506	0.55905	0.39741	0.30161	0.50639
12	1	0.63920	0.70045	0.69731	0.38402	0.34613
13	1	0.10586	0.14880	0.03978	0.08515	0.00402
14	1	0.07787	0.15667	0.03012	0.05951	0.19462
15	1	0.59459	0.65865	0.60413	0.54519	0.22250
16	0.5	0.07398	0.13564	0.03542	0.03565	0.35406
17	1	0.47935	0.70140	0.66480	0.50303	0.35882
18	1	0.27957	0.34890	0.42923	0.11117	0.14207
19	1	0.48151	0.60476	0.51328	0.25837	0.21664
20	1	0.44215	0.70336	0.39795	0.38019	0.15104
21	1	0.13638	0.13337	0.03737	0.04535	0.00090
22	1	0.14685	0.09237	0.00545	0.03272	0.02145
23	0	0.43753	0.50980	0.54906	0.51693	0.54718
24	0	0.80689	0.85628	0.30765	0.56400	0.11189
25	1	0.75464	0.67303	0.60631	0.67779	0.27697
26	1	0.11906	0.14056	0.04816	0.04930	0.00246
27	1	0.80591	0.78415	0.48901	0.60444	0.21484
28	0.5	0.07018	0.14573	0.02869	0.05440	0.18737
29	1	0.08818	0.18531	0.00946	0.01416	0.00004

Lampiran C

Korelasi antar Variabel *Input* dan *Output*



Lampiran D

Coding pada *environment* Spyder pada aplikasi Anaconda Navigator dengan bahasa Python

```
'''##IMPORT LIBRARY##'''
import pandas as pd
import numpy as np
import seaborn as sns

from sklearn.model_selection import cross_val_score
#logistics_libs
from sklearn.linear_model import LogisticRegression
#svm_libs
from sklearn.svm import SVC
#naive_bayes_libs
from sklearn.naive_bayes import GaussianNB
#conf_libs
from sklearn.metrics import confusion_matrix, classification_report

'''### INPUT DATA ###'''
missing_values = ["#DIV/0!","nan"]
data = pd.read_csv("preprocessed data.csv", na_values=missing_values)

'''### DATA CLEANING ###'''

'#visualize data kosong'
#sns.heatmap(data.isnull())
#data[data['dataperuser']=='nan']].sum()

#mean per kolom
vpu_mean = data['voice'].mean()
spu_mean = data['sms'].mean()
dapu_mean = data['data'].mean()
dpu_mean = data['digital'].mean()
payload_mean = data['payload'].mean()

vpu_mean = int(vpu_mean)
spu_mean = int(spu_mean)
dapu_mean = int(dapu_mean)
dpu_mean = int(dpu_mean)
payload_mean = int(payload_mean)

#filling NaN
data['voice'].fillna(vpu_mean, inplace=True)
data['sms'].fillna(spu_mean, inplace=True)
data['data'].fillna(dapu_mean, inplace=True)
data['digital'].fillna(dpu_mean, inplace=True)
data['payload'].fillna(payload_mean, inplace=True)

#Outlier Detection
def find_outlier_tukey (x):
    q1 = np.percentile(x, 25)
    q3 = np.percentile(x, 75)
    iqr = q3-q1
    floor = q1 - 1.5*iqr
    ceiling = q3 + 1.5 * iqr
    outlier_indices = list(x.index[(x<floor)|(x>ceiling)])
    outlier_values = list(x[outlier_indices])
    return (q1, q3, iqr, floor, ceiling, len(outlier_values))

```

Coding pada *environment* Spyder pada aplikasi Anaconda Navigator dengan bahasa Python (Lanjutan)

```
'''#Removing_Outlier'''
## dijalankan per iterasi
##iterasi 1
q1 = np.percentile(data['voice'], 25)
q3 = np.percentile(data['voice'], 75)
iqr = q3-q1
floor = q1 - 1.5*iqr
ceiling = q3 + 1.5 * iqr
voice1 = data[data['voice']>floor]
voice2 = voice1[data['voice']<ceiling]

##iterasi2 - sms
q1 = np.percentile(voice2['sms'], 25)
q3 = np.percentile(voice2['sms'], 75)
iqr = q3-q1
floor = q1 - 1.5*iqr
ceiling = q3 + 1.5 * iqr
sms1 = voice2[voice2['sms']>floor]
sms2 = sms1[sms1['sms']>floor]

##iterasi 3 - dataperuser
q1 = np.percentile(data['sms'], 25)
q3 = np.percentile(data['sms'], 75)
iqr = q3-q1
floor = q1 - 1.5*iqr
ceiling = q3 + 1.5 * iqr
data1 = sms2[sms2['data']>floor]
data2 = data1[data1['data']<ceiling]

##iterasi4 - digiperuser
q1 = np.percentile(data2['digital'], 25)
q3 = np.percentile(data2['digital'], 75)
iqr = q3-q1
floor = q1 - 1.5*iqr
ceiling = q3 + 1.5 * iqr
digi1 = data2[data2['digital']>floor]
digi2 = digi1[digi1['digital']<ceiling]

##iterasi 5 - payload1
q1 = np.percentile(digi2['payload'], 25)
q3 = np.percentile(digi2['payload'], 75)
iqr = q3-q1
floor = q1 - 1.5*iqr
ceiling = q3 + 1.5 * iqr
pload1 = digi2[digi2['payload']>floor]
pload2 = pload1[pload1['payload']<ceiling]

#iterasi 6- payload2
q1 = np.percentile(pload2['payload'], 25)
q3 = np.percentile(pload2['payload'], 75)
iqr = q3-q1
floor = q1 - 1.5*iqr
ceiling = q3 + 1.5 * iqr
pload3 = pload2[pload2['payload']>floor]
pload4 = pload3[pload3['payload']<ceiling]

```

Coding pada *environment* Spyder pada aplikasi Anaconda Navigator dengan bahasa Python (Lanjutan)

```
112
113 #iterasi 7 - payload3
114 q1 = np.percentile(pload4['payload'], 25)
115 q3 = np.percentile(pload4['payload'], 75)
116 iqr = q3-q1
117 floor = q1 - 1.5*iqr
118 ceiling = q3 + 1.5 * iqr
119 pload5 = pload4[pload4['payload']>floor]
120 pload6 = pload5[pload5['payload']<ceiling]
121
122 #iterasi 8 - payload4
123 q1 = np.percentile(pload6['payload'], 25)
124 q3 = np.percentile(pload6['payload'], 75)
125 iqr = q3-q1
126 floor = q1 - 1.5*iqr
127 ceiling = q3 + 1.5 * iqr
128 pload7 = pload6[pload6['payload']>floor]
129 pload8 = pload7[pload7['payload']<ceiling]
130
131 #iterasi 9 - payload5
132 q1 = np.percentile(pload8['payload'], 25)
133 q3 = np.percentile(pload8['payload'], 75)
134 iqr = q3-q1
135 floor = q1 - 1.5*iqr
136 ceiling = q3 + 1.5 * iqr
137 pload9 = pload8[pload8['payload']>floor]
138 pload10 = pload9[pload9['payload']<ceiling]
139
140 #iterasi 10 - digi2
141 ##payloadperuser
142 q1 = np.percentile(pload10['digital'], 25)
143 q3 = np.percentile(pload10['digital'], 75)
144 iqr = q3-q1
145 floor = q1 - 1.5*iqr
146 ceiling = q3 + 1.5 * iqr
147 digi3 = pload8[pload8['digital']>floor]
148 digi4 = digi3[digi3['digital']<ceiling]
149
150 #iterasi 11 - digi3
151 q1 = np.percentile(pload3['digital'], 25)
152 q3 = np.percentile(pload3['digital'], 75)
153 iqr = q3-q1
154 floor = q1 - 1.5*iqr
155 ceiling = q3 + 1.5 * iqr
156 digi5 = digi4[digi4['digital']>floor]
157 digi6 = digi5[digi5['digital']<ceiling]
158
159 #iterasi 12 - digi4
160 q1 = np.percentile(digi6['digital'], 25)
161 q3 = np.percentile(digi6['digital'], 75)
162 iqr = q3-q1
163 floor = q1 - 1.5*iqr
164 ceiling = q3 + 1.5 * iqr
165 digi7 = digi6[digi6['digital']>floor]
166 digi8 = digi7[digi7['digital']<ceiling]
167
```

Coding pada *environment* Spyder pada aplikasi Anaconda Navigator dengan bahasa Python (Lanjutan)

```
#iterasi 13 - digi5
q1 = np.percentile(digi8['digital'], 25)
q3 = np.percentile(digi8['digital'], 75)
iqr = q3-q1
floor = q1 - 1.5*iqr
ceiling = q3 + 1.5 * iqr
digi9 = digi8[digi8['digital']>floor]
digi10 = digi9[digi9['digital']<ceiling]

#iterasi 14 - sms
q1 = np.percentile(digi10['sms'], 25)
q3 = np.percentile(digi10['sms'], 75)
iqr = q3-q1
floor = q1 - 1.5*iqr
ceiling = q3 + 1.5 * iqr
sms3 = digi10[digi10['sms']>floor]
sms4 = sms3[sms3['sms']<ceiling]

#iterasi 15 - payload6
q1 = np.percentile(sms4['payload'], 25)
q3 = np.percentile(sms4['payload'], 75)
iqr = q3-q1
floor = q1 - 1.5*iqr
ceiling = q3 + 1.5 * iqr
pload11 = sms4[sms4['payload']>floor]
pload12 = pload11[pload11['payload']<ceiling]

#iterasi 16 - pload7
q1 = np.percentile(pload12['payload'], 25)
q3 = np.percentile(pload12['payload'], 75)
iqr = q3-q1
floor = q1 - 1.5*iqr
ceiling = q3 + 1.5 * iqr
pload13 = pload12[pload12['payload']>floor]
new_data = pload13[pload13['payload']<ceiling]

#pembagian variabel input (X) dan variabel output (y)
X = new_data.iloc[:,[0,1,2,3,4,5]]
y = new_data.iloc[:,6]

'''# Variable Significance#'''
#menghitung t table
from scipy.stats import t
alpha = 0.05
t.ppf(1-alpha, df = 6782)
#menghitung statistik summary - OLS
import statsmodels.api as sm
X2 = sm.add_constant(X)
est = sm.OLS(y, X2)
est2 = est.fit()
print(est2.summary())

'''Visualisasi Data'''
#korelasi variabel cluster, voice, dan sms
sns.pairplot(new_data.iloc[:,[0,1,2,6]],hue='CHURN',palette='mako')
```

Coding pada *environment* Spyder pada aplikasi Anaconda Navigator dengan bahasa Python (Lanjutan)

```
'''Visualisasi Data'''
#korelasi variabel cluster, voice, dan sms
sns.pairplot(new_data.iloc[:,[0,1,2,6]],hue='CHURN',palette='mako')
#korelasi variabel data, digital, payload
sns.pairplot(new_data.iloc[:,[3,4,5,6]],hue='CHURN',palette='mako')
#korelasi semua variabel input
sns.pairplot(new_data.iloc[:,[0,2,3,4,5,6]],hue='CHURN',palette='mako')


#variabel voice tidak dilibatkan untuk mencegah multikolinearitas
X = X.iloc[:,[0,2,3,4,5]]
#Pembagian data Training dan Data Testing
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)

#Feature scaling
from sklearn.preprocessing import MinMaxScaler
sc_X = MinMaxScaler()
X_test = sc_X.fit_transform(X_test)
X_train = sc_X.transform(X_train)


'''### Logistic Regression ###'''

log_performance = []
cr_log=[]

penalty = ['l1','l2']


for i in penalty:
    #membangun model
        classifier = LogisticRegression(C=1, penalty= i,random_state = 101)
        classifier.fit(X_train,y_train)
        predict1 = classifier.predict(X_test)
        log_cr = classification_report(y_test,predict1)

    #ukur performansi
        log_cr = classification_report(y_test,predict1)
        log_cm = confusion_matrix(y_test,predict1)

        log_acc = cross_val_score(estimator = classifier, X=X, y=y, scoring='accuracy', cv=10)
        log_recal = cross_val_score(estimator=classifier, X=X,y=y, scoring = "recall", cv=10)
        log_prec = cross_val_score(estimator=classifier, X=X, y=y, scoring = "precision", cv=10)
        log_f1 = cross_val_score(estimator=classifier, X=X, y=y, scoring = "f1", cv=10)
        log_me = 1-log_acc

    #rekap performansi
        acc_log = [log_acc.mean(), log_recal.mean(), log_prec.mean(), log_f1.mean(), log_me.mean()]

    #hasil Logistic Regression
        log_performance.append(acc_log)
        cr_log.append(log_cr)

    #mengetahi hasil logistic regression
pd.DataFrame(log_performance)
```

Coding pada *environment* Spyder pada aplikasi Anaconda Navigator dengan bahasa Python (Lanjutan)

```
'''### Support Vector Machine ###'''
svm_kernel = ['linear','rbf','poly'] #i
svm_c = [0.1,1,10,100] #j
svm_rbf_gamma = [0.1,0.25,0.5,0.75] #k
svm_poly_degree = [2,3,5,7]#m

svm_performance = []
cr_svm = []
cm_svm = []
for i in svm_kernel:
    if i == 'linear':
        for j in svm_c:
            #membangun model
            classifier = SVC(C=j, kernel='linear',random_state=101)
            classifier.fit(X_train,y_train)
            predict1 = classifier.predict(X_test)

            #ukur performansi model
            svm_cr = classification_report(y_test,predict1)
            svm_cm = confusion_matrix(y_test,predict1)

            svm_acc = cross_val_score(estimator=classifier,X = X, y=y, cv=10)*r
            svm_recal = cross_val_score(estimator=classifier, X=X, y=y, scoring = "recall", cv=10)*r
            svm_prec = cross_val_score(estimator=classifier, X=X, y=y, scoring = "precision", cv=10)*r
            svm_f1 = cross_val_score(estimator=classifier, X=X, y=y, scoring = "f1", cv=10)*r
            svm_me = 1 - svm_acc

            #rekap performansi
            svm_linear = [i,j,' ', svm_acc.mean(),svm_recal.mean(), svm_prec.mean(), svm_f1.mean(), svm_me.mean()]

            #hasil performansi
            svm_performance.append(svm_linear)
            cr_svm.append(svm_cr)

    elif i == 'rbf':
        for j in svm_c:
            for k in svm_rbf_gamma:
                #membangun model
                classifier = SVC(C=j, kernel='rbf',gamma=k, random_state=101)
                classifier.fit(X_train,y_train)
                predict1 = classifier.predict(X_test)

                #ukur performansi model
                svm_cr = classification_report(y_test,predict1)
                svm_cm = confusion_matrix(y_test,predict1)

                svm_acc = cross_val_score(estimator=classifier,X=X, y=y, cv=10)*r
                svm_recal = cross_val_score(estimator=classifier, X=X, y=y, scoring = "recall", cv=10)*r
                svm_prec = cross_val_score(estimator=classifier, X=X, y=y, scoring = "precision", cv=10)*r
                svm_f1 = cross_val_score(estimator=classifier,X=X, y=y, scoring = "f1", cv=10)*r
                svm_me = 1 - svm_acc

                #rekap performansi
                svm_rbf = [i,j,k,svm_acc.mean(),svm_recal.mean(), svm_prec.mean(), svm_f1.mean(), svm_me.mean()]
                #hasil performansi
                svm_performance.append(svm_rbf)
                cr_svm.append(svm_cr)
```

Permissions: RW End-of-lin

Coding pada *environment* Spyder pada aplikasi Anaconda Navigator dengan bahasa Python (Lanjutan)

```
                cr_svm.append(svm_cr)
    else:
        for j in svm_c:
            for m in svm_poly_degree:

                classifier = SVC(C=j, kernel='poly',degree=m, random_state=101)
                classifier.fit(X_train,y_train)
                predict1 = classifier.predict(X_test)

                #ukur performansi
                svm_cr = classification_report(y_test,predict1)
                svm_cm = confusion_matrix(y_test,predict1)

                svm_acc = cross_val_score(estimator=classifier,X=X, y=y, cv=10)*r
                svm_recal = cross_val_score(estimator=classifier, X=X, y=y, scoring = "recall", cv=10)*r
                svm_prec = cross_val_score(estimator=classifier, X=X, y=y, scoring = "precision", cv=10)*r
                svm_f1 = cross_val_score(estimator=classifier, X=X, y=y, scoring = "f1", cv=10)*r
                svm_me = 1 - svm_acc

                #rekap performansi
                svm_poly = [i,j, m,svm_acc.mean(),svm_recal.mean(), svm_prec.mean(), svm_f1.mean(), svm_me.mean()]

                #hasil performansi
                svm_performance.append(svm_poly)
                cr_svm.append(svm_cr)

#mengetahi hasil support vector machine
pd.DataFrame(log_performance)


'''### Naive_Bayes ####'''
nb_performance = []
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import mean_squared_error

classifier = GaussianNB()
classifier.fit(X_train,y_train)
predict1 = classifier.predict(X_test)

nb_cr = classification_report(predict1, y_test)

nb_acc = cross_val_score(estimator = classifier,X=X, y=y, scoring='accuracy', cv=10)
nb_recal = cross_val_score(estimator=classifier, X=X, y=y, scoring = "recall", cv=10)
nb_prec = cross_val_score(estimator=classifier, X=X, y=y, scoring = "precision", cv=10)
nb_f1 = cross_val_score(estimator=classifier,X=X, y=y, scoring = "f1", cv=10)
nb_me = 1-nb_acc
nb_performance = [nb_acc.mean(), nb_recal.mean(), nb_prec.mean(), nb_f1.mean(),nb_me.mean()]

#mengetahui hasil naive_bayes
pd.DataFrame(nb_performance)
```

Permissions: RW End-of-lines:

(Halaman ini sengaja dikosongkan)

BIOGRAFI PENULIS

Penulis lahir di Pamekasan pada tanggal 10 Oktober 1997 dengan nama lengkap Ach. Nafila Rozie atau biasa dipanggil dengan nama Rozie. Penulis merupakan anak ketiga dari 3 bersaudara. Penulis telah menempuh pendidikan formal di TK Nurul Hikmah Pamekasan, SDN Barurambat Kota I Pamekasan, SMPN 2 Pamekasan, dan SMAN 1 Pamekasan. Pada tahun 2015 penulis diterima sebagai mahasiswa di Departemen Teknik Industri Institut Teknologi Sepuluh Nopember Surabaya.

Selama perkuliahan, penulis aktif dalam beberapa kegiatan organisasi mahasiswa diantaranya sebagai Staf Departemen Sosial Masyarakat di Himpunan Mahasiswa Teknik Industri ITS 16/17 dan juga sebagai Staf Inkubator Kajian BEM ITS 17/18. Selain itu penulis juga mengikuti beberapa pelatihan selama perkuliahan, diantaranya adalah LKMM Pra-TD, LKMM TD, LKMM TM, PP LKMM, Gedhig Manggala Bangsa, dan Pelatihan Pemimpin Bangsa #10 UGM. Penulis juga melakukan beberapa sertifikasi diantaranya *python bootcamp, machine learning,* hingga *SQL*. Kepanitiaan yang dijalani penulis selama perkuliahan antara lain diantaranya fasilitator pada GERIGI ITS 2016, sebagai *Steering Committee* pada SISTEM HMTI ITS 2016 dan GERIGI ITS 2017, Sie Acara pada IE Games 2016 dan Meet the Technocrat BEM FTI 2016, serta Sie Keamanan dan Perizinan pada Inchall 2016. Penulis melaksanakan kegiatan Kerja Praktek di PT Barata Indonesia pada periode Juli-Agustus 2018. Mata kuliah pilihan yang diambil oleh penulis diantaranya adalah: Manajemen Kinerja, Analisa Keputusan, Manajemen Risiko, Data Mining, serta Permodelan Sistem Berbasis Agen. Untuk informasi lebih lanjut mengenai hasil penelitian tugas akhir, penulis dapat dihubungi melalui alamat *e-mail* ach.nafilarozie@gmail.com.

(Halaman ini sengaja dikosongkan