TESIS - SS14 2501

ANALISIS ENSEMBLE SUPPORT VECTOR MACHINE DAN SURVIVAL SUPPORT VECTOR MACHINE PADA DATA NASABAH GADAI DI PERUSAHAAN FINANCIAL TECHNOLOGY - X

Mohammad Alfan Alfian Riyadi
NRP. **06211650010027**

DOSEN PEMBIMBING
Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.
Santi Wulan Purnami, Ph.D.

PROGRAM MAGISTER
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

..

THESIS - SS14 2051

ENSEMBLE SUPPORT VECTOR MACHINE AND SURVIVAL SUPPORT VECTOR MACHINE ANALYSIS IN PAWNING COSTUMER DATA AT FINANCIAL TECHNOLOGY COMPANY-X

Mohammad Alfan Alfian Riyadi
NRP. **06211650010027**

SUPERVISORS
Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.
Santi Wulan Purnami, Ph.D.

PROGRAM OF MAGISTER
DEPARTEMENT OF STATISTICS
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

ANALISIS *ENSEMBLE SUPPORT VECTOR MACHINE* DAN SURVIVAL *SUPPORT VECTOR MACHINE* PADA DATA NASABAH GAIDAI DI PERUSAHAAN *FINANCIAL TECHNOLOGY* - X

Disusun untuk memenuhi syarat memperoleh gelar Magister Sains (M.Si)
di
Institut Teknologi Sepuluh Nopember

Oleh :
MOHAMMAD ALFAN ALFIAN RIYADI
NRP. 06211650010027

Tanggal Ujian : 11 Juli 2018
Periode Wisuda : September 2018

Disetujui Oleh :

1. Dr. rer. pol. Dedy Dwi Prastyo, M.Si (Pembimbing I)
   NIP. 19831204 200812 1 002
2. Santi Wulan Purnami, M.Si, Ph.D (Pembimbing II)
   NIP. 19720923 199803 2 001
3. Dr. rer. pol. Heri Kuswanto, M.Si (Penguji I)
   NIP. 19820326 200312 1 004
4. Dr. Agus Suharsono, M.S (Penguji II)
   NIP. 19580823 198403 1 003

Dekan
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

Prof. Dr. Basuki Widodo, M.Sc
NIP. 19650605 198903 1 002

ANALISIS *ENSEMBLE SUPPORT VECTOR MACHINE* DAN *SURVIVAL SUPPORT VECTOR MACHINE* PADA DATA NASABAH GADAI DI PERUSAHAAN *FINANCIAL TECHNOLOGY* - X

Nama Mahasiswa	: Mohammad Alfan Alfian Riyadi
NRP	: 06211650010027
Dosen Pembimbing	: Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si.
	Santi Wulan Purnami, Ph.D.

ABSTRAK

Terdapat dua kategori nasabah gadai pada perusahaan Fintech X yakni nasabah *early payment* dan *late payment.* Setiap kategori nasabah terdapat durasi pelunasan barang tanggungan. Oleh sebab itu penting bagi perusahaan untuk mendapat informasi awal terkait kondisi nasabah apakah baik atau buruk. Nasabah yang baik adalah nasabah yang semakin cepat dalam melunasi tanggungan sedangkan nasabah yang buruk merupakan nasabah yang semakin lama melunasi tanggungan. Untuk mengatasi problem tersebut terdapat dua tahap permodelan yang dilakukan. Tahap pertama adalah klasifikasi nasabah yang *early payment* atau *late payment*. Tahap kedua menganalisis *survival* untuk masing-masing kategori nasabah. Adapun metode yang digunakan pada tahap pertama yakni Regresi Logistik Biner, SVM dan *Ensemble* SVM. Sedangkan pada tahap kedua adalah *Cox Proportional Hazard* dan *survival* SVM. Untuk mendukung kesimpulan pada tahap klasifikasi, dilakukan studi simulasi dengan membangkitan beberapa skenario variabel prediktor. Hasil studi simulasi diperoleh bahwa *Ensemble* SVM mampu mengimbangi kinerja SVM dan regresi logistik. Akan tetapi ketika diaplikasikan pada data nasabah *Fintech* X, peforma metode klasifikasi yang diajukan tidak memberikan hasil yang baik. Hal tersebut disebabkan tidak adanya variabel yang benar-benar dapat mendiskriminasi kategori nasabah *early payment* maupun *late payment*. Pada tahap berikutnya, *survival* SVM memiliki peforma yang baik dibandingkan *Cox Proportional Hazard*. *Survival* SVM unggul pada setiap kategori nasabah. Salah satu kemungkinan *survival* SVM unggul karena asumsi dari *Cox Proportional Hazard* tidak terpenuhi.

Kata kunci: Analisis *Survival*, Gadai, *Cox Proportional Hazard*, *Ensemble Support Vector Machine*, *Survival Support Vector Machine*.

ENSEMBLE SUPPORT VECTOR MACHINE AND SURVIVAL SUPPORT VECTOR MACHINE ANALYSIS IN PAWNING COSTUMER DATA AT FINANCIAL TECHNOLOGY COMPANY-X

Nama of Student	: Mohammad Alfan Alfian Riyadi
ID	: 06211650010027
Supervisors	: Dr.rer.pol. Dedy Dwi Prastyo, S.Si., M.Si. Santi Wulan Purnami, Ph.D.

ABSTRACT

There are two categories of pawning customers in Fintech X companies, namely early payment and late payment customers. Each category of customer there is the duration of repayment of dependent goods. Therefore it is important for the company to get initial information related to the condition of the customer whether good or bad. A good customer is a customer who is getting faster in paying off the dependents while a bad customer is a customer who is paying off the dependent longer. To overcome the problem there are two stages of modeling. The first stage is the classification of customers who are early payment or late payment. The second phase analyzes survival for each customer category. The method used in the first stage of Binary Logistic Regression, SVM and Ensemble SVM. While in the second stage is Cox Proportional Hazard and SVM survival. To support the conclusions at the classification stage, a simulation study was conducted by generating some predictor variable scenarios. The results of the simulation study found that Ensemble SVM is able to compensate for SVM performance and logistic regression. However, when applied to customer data Fintech X, the performance of the proposed classification method does not give good results. This is due to the absence of variables that can really discriminate the category of early payment customers and late payment. In the next stage, SVM survival has a better performance than Cox Proportional Hazard. SVM Survival excels in every customer category. One possible survival of SVM better because the assumption of Cox Proportional Hazard is not met.

Keywords: Cox Proportional Hazard, Ensemble Support Vector Machine, Pawning, Survival Analysis, Survival Support Vector Machine.

KATA PENGANTAR

Puji Syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik serta hidayah-Nya kepada penulis, sehingga penulis dapat menyelesaikan tesis yang berjudul "ANALISIS ENSEMBLE SUPPORT VECTOR MACHINE DAN SURVIVAL SUPPORT VECTOR MACHINE PADA DATA NASABAH GADAI DI PERUSAHAAN FINANCIAL TECHNOLOGY - X", ini dapat diselesaikan tepat pada waktunya.

Tesis ini diharapkan dapat memberikan manfaat kepada penulis pada khususnya dan pembaca pada umumnya. Penulisan tesis ini belum sempurna (tak ada gading yang tak retak), karena kesempurnaan hanya milik sang Pencipta, oleh karena itu penulisan mengharapkan kritik dan saran bagi pembaca untuk mendapatkan hasil yang lebih baik di kemudian hari.

Penulisan tesis ini tidak akan berjalan dengan lancar tanpa bantuan dan dukungan beberapa pihak, oleh karena itu pada kesempatan kali ini penulis ingin mengucapkan terimakasih kepada:

1. Kedua orang tua dan keluarga penulis yang telah memberikan dukungan baik secara moril dan materiil kepada penulis sehingga penulis dapat menyelesaikan dengan baik.
2. Bapak Dr. Suhartono, M.Sc., selaku Kepala Departemen Statistika FMKSD ITS.
3. Bapak Dr. rer. pol. Dedy Dwi Prasetyo, S.Si. M.Si., selaku pembimbing utama yang telah membimbing penulis hingga dengan penuh kesabaran, memberikan masukan, serta arahan demi terselesaikannya tesis ini.
4. Ibu Santi Wulan Purnami, M.Si., Ph.D., selaku pembimbing kedua yang telah memberikan saran, masukan serta arahan yang membangun demi kesempurnaan tesis ini.
5. Bapak Dr. rer. pol. Heri Kuswanto, S.Si, M.Si. dan Bapak Dr. Agus Suharsono, M.S., selaku penguji yang memberikan masukan dan arahan pada penyelesaian tesis ini.

6. Seluruh Dosen Statistika ITS yang telah membekali ilmu kepada penulis sehingga dapat menyelesaikan tesis ini.
7. Rekan-rekan Data Science Indonesia Chapter East Java, Mas Sofian Hadiwijaya, dan Mas Risky yang sudah menyempatkan untuk belajar Bersama.
8. Teman-teman seperjuangan S2 Statistika ITS angkatan 2016 yang telah berjuang bersama-sama untuk menyelesaikan studi magister.
9. Semua teman, relasi, dan berbagai pihak yang tidak bisa penulis sebutkan namanya satu per satu yang telah membantu dalam penulisan laporan ini.

Penulis menyadari bahwa tesis ini masih jauh dari sempurna. Oleh karena itu, kritik dan saran diharapkan dari semua pihak untuk tahap pengembangan selanjutnya. Besar harapan penulis bahwa informasi sekecil apapun dalam tesis ini bermanfaat bagi semua pihak dan dapat menambah wawasan serta pengetahuan.

Surabaya, Juli 2018

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

DAFTAR LAMPIRAN

BAB 1
PENDAHULUAN

1.1 Latar Belakang

Gadai merupakan proses peminjaman uang (kredit) dalam batas waktu tertentu dengan memberikan barang tanggungan (biasanya berupa emas, kendaraan bermotor, dan barang berharga lainnya), yang apabila telah sampai pada waktunya barang tidak ditebus maka, maka barang itu menjadi hak pemeberi pinjaman (KBBI, 2017). Semakin berharga suatu barang maka semakin besar pula kredit yang diperoleh. Saat melakukan pelunasan (penebusan), kategori nasabah gadai terbagi menjadi dua macam yakni nasabah yang cenderung membayar sebelum jatuh tempo (*early payment*) dan nasabah yang cenderung membayar terlambat (*late payment*). Kategori nasabah *early payment* maupun *late payment* masing-masing memiliki durasi dalam melakukan pelunasan. Berdasarkan kondisi tersebut, terdapat dua hal terpenting yakni bagaimana kecenderungan nasabah gadai dalam melunasi kreditnya dan berapa lama nasabah melakukan *early payment* maupun *late payment*. Metode statistika yang dapat membantu menyelesaikan permasalahan penentuan kategorisaasi nasabah yakni menggunakan analisis klasifikasi. Sedangkan metode statistika yang dapat digunakan untuk mengukur seberapa lama nasabah melakukan *early payment* dan *late payment* adalah analisis *survival*.

Analisis klasifikasi merupakan analisis yang tujuannya mengidentifikasi seperangkat kategori dari pengamatan baru, berdasarkan kumpulan *data training* yang berisi pengamatan dimana keanggotaan kategorinya diketahui. Analisis klasifikasi pada kasus kredit disebut dengan *credit scoring*. *Credit scoring* serangkaian model keputusan dan teknik dasar yang membantu pemberi pinjaman dalam pemberian kredit konsumen. Teknik ini menentukan siapa yang akan mendapatkan kredit, berapa jumlahnya kredit yang harus mereka dapatkan, dan strategi operasional apa yang akan meningkatkan profitabilitas peminjam ke pemberi pinjaman (Thomas et. al, 1997).

Penelitian mengenai *credit scoring* beberapa tahun ini pernah dilakukan oleh Constangioara (2011) dengan membandingkan metode parametrik (regresi

logistik) dan non parametrik (*decision tree* dan *bagging decision tree*). Adapun variabel penelitian yang digunakan adalah usia, pendidikan, pendapatan, jenis pekerjaan, status pernikahan, besaran pinjaman, dan banyaknya anggota keluarga. Diperoleh hasil bahwa variabel tersebut signifikan dalam menentukan *credit scoring.* Selain itu, metode non parametrik menghasilkan peforma yang lebih baik. Ghodselahi (2011) membandingkan metode *Ensemble* Support Vector Machine (SVM) dengan beberapa metode parametrik dan non parametrik. Pada penelitian tersebut diperoleh hasil bahwa metode *Ensemble* SVM memberikan peforma yang lebih baik. Han et al (2012) membandingkan regresi logistik dengan SVM. Data yang digunakan adalah Germany Dataset. Diperoleh hasil bahwa SVM memberikan performa yang lebih baik dibandingkan regresi logistik.

Analisis *survival* merupakan prosedur statistik untuk data analisis dengan *ouput* variabel berupa waktu hingga suatu kejadian (*event*) terjadi. Waktu kejadian dapat diukur dalam hari, minggu, bulan, tahun saat suatu pengamatan diamati hingga terjadinya suatu kejadian. Sedangkan *event* yang dimaksud adalah setiap pengalaman dari suatu observasi yang kemungkinan akan terjadi (Kleinbaum dan Klein, 2012). Dua hal yang paling fundamental dari analisis *survival* adalah *survival function* dan *hazard function*. *Survival function* merupakan probabilitas suatu objek *survive* pada waktu tertentu. *Hazard function* menyatakan laju *failure* suatu objek (Moore, 2016).

Penelitian mengenai analisis *survival* pada kasus kredit personal pertama kali dilakukan oleh Narain (1992) dengan membuat *credit scoring* untuk nasabah dengan mengestimasi waktu nasabah tidak dapat membayar kredit (*default*) atau yang *early payment*. Kemudian Thomas *et al* (1999). melakukan analisis survival terhadap 50.000 nasabah yang mengajukan kredit pada tahun 1994 hingga 1997. Metode yang digunakan adalah membandingkan regresi logistik dan beberapa metode analisis survival paramterik maupun semi parametrik seperti *Weibull*, *Exponential*, dan *Cox Proportional Hazard*. Diperoleh hasil bahwa metode analisis *survival* mampu menjelaskan waktu *default* nasabah kredit maupun *early payment.* Stephanova dan Thomas (2000) melanjutkan penelitian tersebut dengan menambahkan karakteristik alasan nasabah mengajukan kredit. Sebagai catatan bahwa terdapat kelemahan pada analisis *survival* parametrik maupun semi

parametrik yakni jika beberapa asumsi tidak terpenuhi, terjadinya interaksi antar variabel, dll. Sehingga untuk menjawab permasalahan tersebut Baesens *et al.* (2005) melakukan analisis *survival* menggunakan metode non parametrik yakni *neural network*.

Perkembangan analisis survival dengan metode non parametrik tidak hanya sampai disitu saja. Van Belle *et al.* (2007) dan (2008) menerapkan analisis *survival* dengan pendekatan *Support Vector Machine* (SVM). Van Belle *et al.* (2008) menggunakan *survival* SVM untuk membandingkan performansi metode dengan menggunakan *Cox Proportional Hazard*, *Accelerated Failure Time Model* (AFT model), cSVM (SVM dengan linear kernel), dan cSVM- Gaussian Radial Basis (SVM dengan RBF kernel). Dengan menggunakan dua data set, didapatkan kesimpulan bahwa hasil dengan menggunakan *survival* SVM lebih baik dari pada menggunakan *Cox Proportional Hazard* maupun AFT. Selain itu Khotimah *et. al.* (2017) dan (2018) juga membandingkan antara *Cox Proportional Hazard* dan *Survival* SVM. Diperoleh hasil bahwa metode *Survival* SVM lebih baik dibandingkan dengan *Cox Proportional Hazard*.

Berdasarkan penjelasan yang telah dipaparkan, pada penelitian ini akan menggunakan model parametrik dan non parametik. Hal ini disebabkan ada kemungkinan model yang telah dibangun tidak memenuhi asumsi. Proses analisis pada penelitian ini terdapat dua tahap yakni tahapan pertama adalah klasifikasi nasabah yang *early payment* atau terlambat melunasi kredit. Tahap kedua menganalisis *survival* untuk masing-masing kategori nasabah. Adapun metode yang digunakan pada tahap pertama yakni Regresi Logistik Biner, SVM dan *Ensemble* SVM. Sedangkan pada tahap kedua adalah *Cox Proportional Hazard* dan *survival* SVM. Data yang digunakan pada penelitian ini adalah data nasabah gadai perusahaan *financial technology* (*FinTech*) X.

1.2 Rumusan Masalah

Rumusan masalah berdasarkan permasalahan di atas adalah pada kasus data nasabah gadai *fintech* X terdapat klasifikasi antara nasabah *early payment* dan *late payment*. Setiap kondisi nasabah baik *early* maupun *late* terdapat durasi pelunasan. Oleh sebab itu terdapat dua permasalahan sekaligus yakni permasalahan klasifikasi dan survival. Pada permasalahan klasifikasi ingin diketahui metode mana yang

lebih baik untuk menglasifikasikan antara nasabah *early payment* dengan *late payment*. Metode yang digunakan untuk klasifikasi adalah Regresi Logistik, SVM, dan *Ensemble* SVM. Namun pada proses klasifikasi dilakukan terlebih dahulu kajian simulasi untuk mendapatkan kesimpulan yang lebih komprehensif. Setelah memperoleh model klasifikasi yang terbaik, metode analisis survival mana yang memiliki performa lebih baik untuk mengukur durasi nasabah *early payment* maupun *late payment*. Metode analisis survival yang digunakan adalah *Cox Proportional Hazard* dan *Survival* SVM..

1.3 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut

1. Melakukan studi simulasi untuk membandingkan performa Regresi Logistik Biner, SVM dan *Ensemble* SVM.
2. Mendapatkan metode klasifikasi terbaik untuk menglasifikasikan antara nasabah *early payment* dengan nasabah *late payment* dengan regresi logistic, SVM, dan *Ensemble SVM*.
3. Memperoleh metode analisis *survival* terbaik antara SVM dengan *Cox Proportional Hazard*

1.4 Manfaat

Penelitian ini diharapkan dapat memperoleh manfaat sebagai berikut

1. Memperoleh pengetahuan terkait pengembangan *ensemble* SVM.
2. Memperoleh pengetahuan terkait pengembangan *survival* SVM pada kasus gadai nasabah.
3. Sebagai rekomendasi perusahaan *FinTech* X untuk mempertimbangkan nasabah yang melakukan gadai dengan memanfaatkan model *machine learning* yang telah dibuat.

1.5 Batasan Masalah

Batasan masalah yang digunakan pada penelitian ini sebagai berikut.

1. Variabel prediktor yang digunakan sebagai penelitian hanya terdapat enam variabel. Hal tersebut dikarenakan perusahaan hanya bisa menyediakan variabel tersebut saja.
2. Fungsi kernel yang digunakan adalah *Radial Basis Function.*

3. Pada prosedur *ensemble* SVM, pada perhitungan klaster *k-means*, variabel kategorik dipaksakan digunakan pada perhitungan jarak *Euclidean*. Tidak ada penanganan khusus pada variabel kategorik.

BAB 2
TINJAUAN PUSTAKA

2.1 Regresi Logistik Biner

Model regresi logistik biner merupakan model matematika yang dapat digunakan untuk memodelkan hubungan antara variabel prediktor X dengan variabel respon *y* yang bersifat biner (Hosmer dan Lemeshow, 2013). Variabel respon (*y*) mengikuti distribusi Bernoulli dengan fungsi probabilitas :

$$f(y_i) = \pi^{y_i}(1-\pi)^{1-y_i}, y_i = 0,1. \quad (2.1)$$

dimana π merupakan probabilitas sukses. Bentuk model regresi logistik adalah (Agresti,2007) :

$$\pi(x) = \frac{\exp(\beta_0 + \beta_1 x)}{1 + \exp(\beta_0 + \beta_1 x)} \quad (2.2)$$

Model regresi logistik dengan *k* variabel prediktor adalah:

$$\pi(\boldsymbol{x}) = \frac{\exp(\beta_0 + \beta_1 x_1 + \ ... + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \ ... + \beta_k x_k)} \quad (2.3)$$

Jika model ditransformasikan dengan transformasi logit, maka akan menghasilkan bentuk logit :

$$g(\boldsymbol{x}) = \ln\left[\frac{\pi(\boldsymbol{x})}{1-\pi(\boldsymbol{x})}\right] = \beta_0 + \beta_1 x_1 + \ ... + \beta_k x_k \quad (2.4)$$

regresi logistik pada Persamaan (2.4) dapat dituliskan dalam bentuk

$$\pi(\boldsymbol{x}) = \frac{\mathrm{e}^{\,g(\boldsymbol{x})}}{1 + \mathrm{e}^{\,g(\boldsymbol{x})}} \quad (2.5)$$

untuk *i* = 1,2,…,n maka model regresi logistik dapat ditulis

$$\pi(\boldsymbol{x}_i) = \frac{\mathrm{e}^{\sum_{j=0}^{k}\beta_j x_{ij}}}{1 + \mathrm{e}^{\sum_{j=0}^{k}\beta_j x_{ij}}}. \quad (2.6)$$

Estimasi parameter dalam regresi logistik dilakukan dengan metode *Maximum Likelihood*. Metode tersebut mengestimasi parameter **β** dengan cara memaksimumkan fungsi *likelihood*. Pada regresi logistik, setiap pengamatan mengikuti Distribusi Bernoulli sehingga dapat ditentukan fungsi *likelihood*nya.

$$f(y_i) = \pi(\boldsymbol{x}_i)^{y_i}(1-\pi(\boldsymbol{x}_i))^{1-y_i}, y_i = 0,1. \quad (2.7)$$

Jika X_i dan Y_i adalah pasangan variabel respon dan prediktor pada pengamatan ke-i yang diasumsikan bahwa setiap pasangan pengamatan saling independen dengan pasangan pengamatan lainnya, maka fungsi *likelihood* merupakan gabungan dari fungsi distribusi masing-masing pasangan yaitu:

$$l(\boldsymbol{\beta}) = \prod_{i=1}^{n} f(\mathrm{y}_i) = \prod_{i=1}^{n} \pi(\mathbf{x}_i)^{\mathrm{y}_i} \left(1-\pi(\mathbf{x}_i)\right)^{1-y_i}$$

$$= \left\{ \prod_{i=1}^{n} \exp\left[\ln\left(\frac{\pi(\mathbf{x}_i)}{1-\pi(\mathbf{x}_i)} \right)^{y_i} \right] \right\} \left\{ \prod_{i=1}^{n} \left[1-\pi(\mathbf{x}_i)\right] \right\}$$

$$= \left\{ \exp\left[\sum_{i=1}^{n} y_i \ln\left(\frac{\pi(\mathbf{x}_i)}{1-\pi(\mathbf{x}_i)} \right) \right] \right\} \left\{ \prod_{i=1}^{n} \left(1-\pi(\mathbf{x}_i)\right) \right\}$$

$$= \left\{ \exp\left[\sum_{i=1}^{n} y_i \ln\left(\frac{\pi(\mathbf{x}_i)}{1-\pi(\mathbf{x}_i)} \right) \right] \right\} \left\{ \prod_{i=1}^{n} \frac{1}{1+\exp\left(\sum_{j=1}^{k} \beta_j x_{ij} \right)} \right\}$$

$$l(\boldsymbol{\beta}) = \left\{ \exp\left(\sum_{i=1}^{n} y_i \sum_{j=1}^{k} \beta_j x_{ij} \right) \right\} \left\{ \prod_{i=1}^{n} \left[1+\exp\left(\sum_{j=1}^{k} \beta_j x_{ij} \right) \right]^{-1} \right\} \qquad (2.8)$$

Fungsi *likelihood* tersebut kemudian dimaksimumkan dalam bentuk ln $l(\boldsymbol{\beta})$ dan dinyatakan dengan $L(\boldsymbol{\beta})$.

$L(\boldsymbol{\beta})$ = ln $l(\boldsymbol{\beta})$

$$L(\boldsymbol{\beta}) = \sum_{j=0}^{k} \left[\sum_{i=1}^{n} y_i x_{ij} \right] \beta_j - \sum_{i=1}^{n} \ln\left[1+\exp\left(\sum_{j=0}^{k} \beta_j x_{ij} \right) \right] \qquad (2.9)$$

Nilai $\boldsymbol{\beta}$ maksimum didapatkan melalui turunan $L(\boldsymbol{\beta})$ terhadap $\boldsymbol{\beta}$ dan hasilnya adalah sama dengan nol.

$$\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{n} y_i x_{ij} - \sum_{i=1}^{n} x_{ij} \left(\frac{\exp\left(\sum_{j=0}^{k} \beta_j x_{ij} \right)}{1+\exp\left(\sum_{j=0}^{k} \beta_j x_{ij} \right)} \right) = 0 \qquad (2.10)$$

sehingga,

$$\sum_{i=1}^{n} y_i x_{ij} - \sum_{i=1}^{n} x_{ij} \hat{\pi}(\mathbf{x}_i) = 0,\ j = 0,1,2,...,k \qquad (2.11)$$

Namun tidak diperoleh hasil yang eksplisit dari Persamaan (2.10). Oleh karena itu diperlukan metode numerik untuk memperoleh estimasi parameternya. Metode iterasi *Newton Raphson* digunakan untuk menyelesaikan persamaan yang non linear.

$$\boldsymbol{\beta}^{(t+1)} = \boldsymbol{\beta}^{(t)} - \left(\mathbf{H}(\boldsymbol{\beta}^{(t)})\right)^{-1} \mathbf{g}(\boldsymbol{\beta}^{(t)}), t = 0,1,2,... \quad (2.12)$$

dengan $\mathbf{g}^T = \left(\frac{\partial L(\boldsymbol{\beta})}{\partial \beta_0}, \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_1}, ..., \frac{\partial L(\boldsymbol{\beta})}{\partial \beta_k} \right)$ dan $\mathbf{H}$ merupakan matriks Hessian dengan elemennya adalah $h_{ju} = \frac{\partial^2 L(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_u}$.

Langkah-langkah iterasi Newton Raphson adalah sebagai berikut:

1. Menentukan nilai awal estimasi parameter $\widehat{\boldsymbol{\beta}}^{(0)}$.
2. Membentuk vektor gradien g dan matriks Hessian H.
3. Memasukkan nilai $\widehat{\boldsymbol{\beta}}^{(0)}$ pada elemen **g** dan **H** sehingga diperoleh $\mathbf{g}(\widehat{\boldsymbol{\beta}}^{(0)})$ dan $H(\widehat{\boldsymbol{\beta}}^{(0)})$.
4. Iterasi mulai t = 0 menggunakan Persamaan (2.12). Nilai $\widehat{\boldsymbol{\beta}}^{(t)}$ merupakan sekumpulan penaksir parameter yang konvergen pada iterasi ke-*t*.
5. Apabila belum diperoleh estimasi parameter yang konvergen, maka langkah (3) diulang kembali hingga nilai $\|\widehat{\boldsymbol{\beta}}^{(t+1)} - \widehat{\boldsymbol{\beta}}^{(t)}\| \leq \varepsilon$, dengan ε merupakan bilangan yang sangat kecil. Hasil estimasi yang diperoleh adalah $\widehat{\boldsymbol{\beta}}^{(t+1)}$ pada iterasi terakhir.

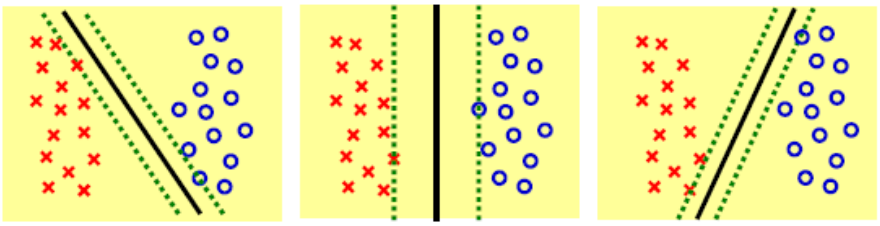
2.2 *Support Vector Machine* (SVM)

SVM adalah suatu sistem pembelajaran dengan menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur (*feature space*) berdimensi tinggi. Konsep SVM menggunakan hyperplane tunggal pada ruang berdimensi banyak yang pada akhirnya partisi-partisi tersebut dapat diselesaikan secara non linier (Gunn,1998).

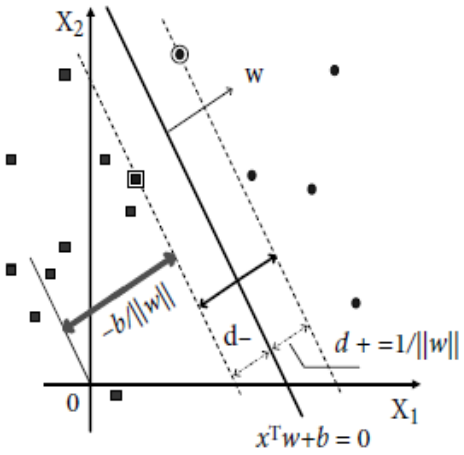
SVM pertama kali dikenalkan oleh Boser, Guyon dan Vapnik (1992). SVM adalah suatu teknik baru untuk melakukan prediksi, baik dalam kasus klasifikasi maupun regresi. SVM berada dalam satu kelas dengan *Neural Network* (NN) dalam

hal fungsi dan kondisi permasalahan yang bisa diselesaikan, keduanya masuk pada kelas *supervised learning*.

SVM secara teoritis dikembangkan untuk masalah klasifikasi dengan dua kelas sebagai usaha mencari *hyperplane* terbaik. *Hyperplane* merupakan fungsi pemisah antara dua kelas pada *input space*. Pada Gambar 2.1 diperlihatkan beberapa data yang merupakan anggota dua kelas, yaitu kelas untuk kategori A dan kategori B. Kelas A dinotasikan dengan +1 (Biru) dan kelas B dinotasikan dengan -1 (Merah). Data yang termasuk dalam kelas perempuan disimbolkan dengan lingkaran. Proses pembelajaran problem klasifikasi diterjemahkan untuk menemukan garis *hyperplane* yang memisahkan antara dua kelas tersebut. Pada Gambar 2.1 (a,c) terlihat alternatif garis pemisah (*discrimination boundaries*), sedangkan pada Gambar 2.1 (b) ditunjukkan bahwa terdapat garis *hyperplane* yang tepat berada diantara dua kelas. Prinsip dasar dari analisis ini adalah menemukan hyperplane terbaik dengan meminimalkan kesalahan klasifikasi dan memaksimalkan margin geometriknya seperti pada Gambar 1 (b) (Prasetyo, 2014).



Gambar 2.1 Ilustrasi Mencari Hyperplane Terbaik



Gambar 2.2 Konsep *Hyperplane* pada SVM (Haerdle, *et.al.*, 2014)

Fungsi klasifikasi $\mathbf{x}^T\mathbf{w}+b$ berada dalam sebuah keluarga fungsi klasifikasi $\mathcal{F}$ yang terbentuk yaitu $\mathbf{x}^T\mathbf{w}+b, \mathbf{w}\in\mathbb{R}^P, b\in\mathbb{R}$. Bidang pemisah (*separating hyperplane*) :

$$f(x)=\boldsymbol{x}^T\boldsymbol{w}+b=0 \tag{2.13}$$

Fungsi pemisah untuk kedua kelas adalah sebagai berikut.

$$\boldsymbol{x}_i^T\boldsymbol{w}+b\geq 1 \text{ untuk } y_i=+1 \tag{2.14}$$

$$\boldsymbol{x}_i^T\boldsymbol{w}+b\leq -1 \text{ untuk } y_i=-1 \tag{2.15}$$

Dimana $\boldsymbol{w}$ adalah vektor bobot (*weight vector*) yang berukuran (*p*x1), *b* adalah posisi bidang relatif terhadap pusat koordinat atau lebih dikenal dengan bias yang bernilai skalar. Pada Gambar 2.2 menunjukkan $\frac{|b|}{\|\boldsymbol{w}\|}$ adalah jarak bidang pemisah yang tegak lurus dari titik pusat koordinat dan $\|\boldsymbol{w}\|$ adalah jarak *euclidian* dari $\boldsymbol{w}$. Bidang batas pertama membatasi kelas (+1) sedangkan bidang pembatas kedua membatasi kelas (-1). *Hyperplane* yang optimal adalah max $\frac{2}{\|\boldsymbol{w}\|}$atau *equvalent* dengan min $\frac{1}{2}\|\boldsymbol{w}\|$. Dengan menggabungkan kedua konstrain pada persamaan (2.14) dan (2.15) maka dapat direpresentasikan dalam pertidaksamaan sebagai berikut.

$$y_i(\boldsymbol{x}_i^T\boldsymbol{w}+b)-1\geq 0, \quad i=1,2,...,n \tag{2.16}$$

Secara matematis, formulasi problem optimasi SVM untuk klasifikasi linier dalam *primal space* adalah

$$\min \tfrac{1}{2}\|\boldsymbol{w}\|^2 \tag{2.17}$$

dengan fungsi kendala $y_i(\mathbf{x}_i^T\mathbf{w}+b)-1\geq 0, \quad i=1,2,...,n$. Secara umum, persoalan optimasi (2.17), akan lebih mudah diselesaikan jika diubah ke dalam formula *lagrange*. Dengan demikian permasalahan optimasi dengan konstrain dapat dirumuskan menjadi

$$L_{\text{pri}}(\boldsymbol{w},b,\alpha)=\frac{1}{2}\|\boldsymbol{w}\|^2-\sum_{i=1}^{n}\alpha_i\left\{y_i(\boldsymbol{x}_i^T\boldsymbol{w}+b)-1\right\}, \tag{2.18}$$

dengan $\alpha_i\geq 0$ (Haerdle, *et al.*, 2014). Namun untuk kasus kasus linearly non-separable perlu ditambahkan batasan berupa variabel slack $\boldsymbol{\xi_i}$ yang menunjukkan

pelanggaran dari pemisahan yang ketat. Sehingga formulasi lagrange pada persamaan 2.18 berubah menjadi.

$$L_P(w,b,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left\{y_i(x_i^T w + b) - 1 + \xi_i\right\} - \sum_{i=1}^{n}\mu_i\xi_i\,, \tag{2.19}$$

dengan $\alpha_i \geq 0$ dan $\mu_i \geq 0$. Permasalahan primal diatas dapat diselesaikan dengan memperoleh turunan pertama untuk masing-masing w, b, dan ξ. Sehingga diperoleh *dual problem* sebagai berikut

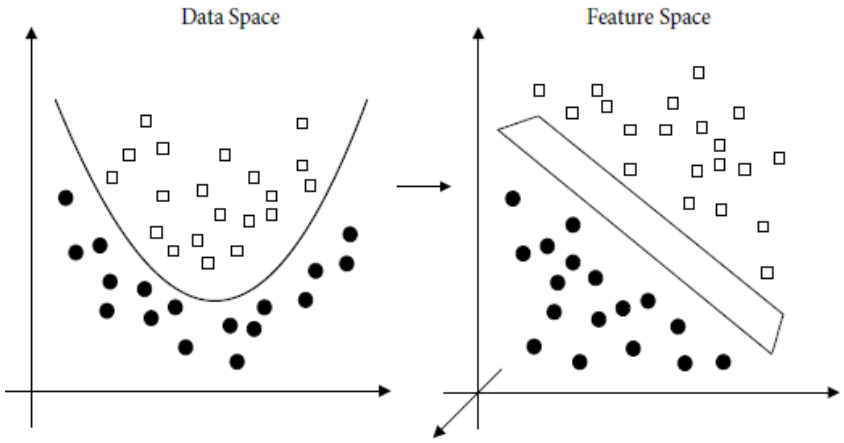
$$L_D(\alpha) = \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j x_i^T x_j\,, \tag{2.20}$$

di mana

$$0 \leq \alpha_i \leq C,$$

$$\sum_{i=1}^{n}\alpha_i y_i = 0$$

Pada dasarnya, SVM merupakan linier *classifier* tetapi seiring berkembangnya penelitian SVM dapat bekerja pada masalah non linier. Pada SVM non linier, fungsi transformasi yang digunakan adalah "*Kernel Trick*" (Scholkopf & Smola, 2002). Penggunaan kernel bertujuan untuk mengimplementasikan suatu model pada ruang dimesi yang lebih tinggi (*feature space*) sehingga kasus yang *non lineary separable* pada ruang input bisa ditransformasi menjadi *lineary separable* pada *feature space*. *Kernel Trick* menghitung *scalar product* dalam bentuk sebuah fungsi kernel.



Gambar 2.3 *Mapping* dari Dua Dimensi Data *Space* (Kiri) ke Tiga Dimensi *Feature Space* (Kanan) (Haerdle *et al.*, 2014)

Diberikan sebuah kernel K dan data $x_1, x_2, x_3, ..., x_n \in \boldsymbol{x}$ maka matrik $K = (K(\boldsymbol{x}_i, \boldsymbol{x}_j))_{ij}$ berukuran nxn disebut *Gram matrix* untuk data $x_1, x_2, x_3, ..., x_n$. Sebuah syarat cukup dan perlu untuk matrik simetri K dengan $K_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j) = K(\boldsymbol{x}_j, \boldsymbol{x}_i) = K_{ji}$, untuk K definit positif disebut *Mercer's Theorem* (Mercer, 1909).

$$\sum_{i=1}^{n}\sum_{j=1}^{n}\varphi_i\varphi_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \geq 0 \tag{2.21}$$

Contoh sederhana pada sebuah *kernel trick* yang menunjukkan bahwa kernel dapat dihitung tanpa perhitungan fungsi *mapping* φ secara eksplisit adalah fungsi pemetaan :

$$\varphi(x_1, x_2) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)^T$$

sehingga menjadi

$$\mathbf{w}^T\varphi(\boldsymbol{x}) = w_1x_1^2 + \sqrt{2}w_2x_1x_2 + w_3x_2^2$$

dengan dimensi pada *feature space* adalah kuadratik, padahal dimensi asalnya adalah linier. Metode kernel menghindari pembelajaran secara eksplisit *mapping* data ke dalam *feature space* dimensi tinggi, sepeti pada contoh berikut.

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{w}^T\boldsymbol{x} + b \\ &= \sum_{i=1}^{n}\alpha_i \boldsymbol{x}_i{}^T\boldsymbol{x} + b \\ &= \sum_{i=1}^{n}\alpha_i\varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}) + b \\ &= \sum_{i=1}^{n}\alpha_i\ K(\boldsymbol{x}_i, \boldsymbol{x}) + b \end{aligned}$$

Hubungan kernel dengan fungsi *mapping* adalah sebagai berikut.

$$\begin{aligned} \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}) &= (x_{i1}^2, \sqrt{2}x_{i1}x_{i2}, x_{i2}^2)(x_1^2, \sqrt{2}x_1x_2, x_2^2)^T \\ &= x_{i1}^2x_1^2 + 2x_{i1}x_{i2}x_1x_2 + x_{i2}^2x_2^2 \\ &= (\boldsymbol{x}_i{}^T\boldsymbol{x})^2 \\ &= K(\boldsymbol{x}_i, \boldsymbol{x}) \end{aligned}$$

Menurut Hsu *et al*. (2010), terdapat 4 fungsi kernel yaitu

1. Kernel Linier

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{2.22}$$

2. Kernel Polynomial

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\mu \boldsymbol{x}_i^T \boldsymbol{x}_j + r)^d, \mu > 0 \tag{2.23}$$

3. Fungsi Kernel Radial Basis Function (RBF)

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|^2}{2\sigma^2} \right) \tag{2.24}$$

4. Kernel Eksponensial

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(\mu \boldsymbol{x}_i^T \boldsymbol{x}_j + r) \tag{2.25}$$

dengan μ, r, d dan σ merupakan parameter kernel dan i, j=1,2,...,m.

Pemilihan fungsi kernel yang tepat merupakan hal yang sangat penting karena akan menentukan *feature space* dimana fungsi *classifier* akan dicari. Sepanjang fungsi kernelnya sesuai (cocok), SVM akan beroperasi secara benar meskipun tidak tahu pemetaan yang digunakan (Santosa, 2007; Robandi, 2008). Menurut Scholkopf dan Smola (2002), fungsi kernel gaussian RBF memiliki kelebihan yaitu secara otomatis menentukan nilai, lokasi dari *center* dan nilai pembobot dan bisa mecakup nilai rentang tak terhingga. Gaussian RBF juga efektif menghindari *overfitting* dengan memilih nilai yang tepat untuk parameter C dan σ dan RBF baik digunakan ketika tidak ada pengetahuan terdahulu. Menurut Hsu, Chang dan Lin (2004), fungsi kernel yang direkomendasikan untuk diuji pertama kali adalah fungsi kernel RBF karena dapat memetakan hubungan tidak linier RBF lebih robust terhadap *outlier* karena fungsi kernel RBF berada antara selang $(-\infty, \infty)$ sedangkan fungsi kernel yang lain memiliki rentang antara (-1 sampai dengan 1).

2.3 Klaster *K-Means* dan *Kernel K-Means*

Klaster *K-Means* merupakan rumpun klaster non hirarki, dimana terdapat proses untuk menentukan terlebih dahulu berapa banyak klaster yang ingin

dibentuk (Johnson dan Wichern, 2014). Proses algoritma klaster *k-means* adalah sebagai berikut:

1. Memilih *k* buah *centroid* secara acak.
2. Mengelompokkan data sehingga terbentuk *k* buah Klaster dengan titik *centroid* dari setiap klaster merupakan titik *centroid* yang telah dipilih sebelumnya. Pada proses ini menghitung jarak antara observasi dengan titik *centroid* yang telah ditentukan. Jarak yang umum digunakan adalah jarak *Euclidean*.
3. Perbaharui nilai titik *centroid.*
4. Ulangi langkah 2 dan 3 sampai nilai dari titik *centroid* tidak lagi berubah.

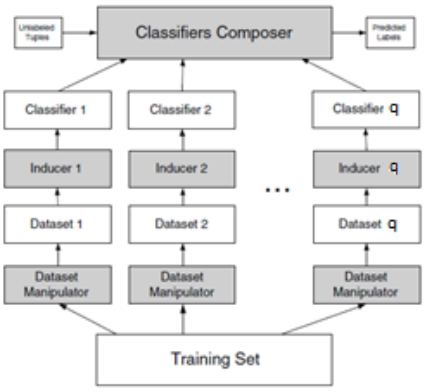
Namun metode *k-means* memiliki kelemahan yakni tidak mampu mengatasi kasus data yang *non* linier. Salah satu metode klaster yang dapat digunakan adalah *kernel k-means*. Ide dari metode ini adalah melakukan transformasi data menuju dimensi yang lebih tinggi dengan menggunakan fungsi mapping φ (Dhilon et.al, 2005). Jika terdapat vector observasi $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n$, maka jarak observasi dengan *centroid* ke j ($\boldsymbol{m}_j$) dimana $j = 1,2, \dots, q$ dinyatakan sebagai berikut

$$\mathcal{D}\left(\varphi(\boldsymbol{x}_i), \boldsymbol{m}_j\right) = \sqrt{\left(\varphi(\boldsymbol{x}_i) - \boldsymbol{m}_j\right)^T \left(\varphi(\boldsymbol{x}_i) - \boldsymbol{m}_j\right)} \tag{2.26}$$

2.3 *Ensemble Support Vector Machine* (SVM)

Ensemble pada kasus klasifikasi merupakan penggabungkan beberapa model klasifikasi dengan harapan dapat meningkatkan performansi dari klasifikasi. Secara umum struktur *ensemble* terbagi menjadi beberapa hal yaitu.

1. *Training set*: Dataset berlabel yang digunakan untuk melatih *ensemble*.
2. *Base Inducer*: Algoritma induksi yang memperoleh *training set* dan membentuk *classifier* yang mewakili hubungan antara atribut *input* dan atribut *output*.
3. *Diversity Generator* berperan untuk menghasilkan beragam klasifikasi.
4. *Combiner* berperan untuk menggabungkan hasil klasifikasi dari berbagai model klasifikasi.



Gambar 2.4 Stuktur Umum Model *Ensemble* (Rokach, 2009)

Salah satu algoritma *ensemble* SVM yang dikembangkan adalah berbasis *cluster*. Algoritma ini dikenalkan oleh Gu dan Han (2013). Diberikan sampel berupa $S = \{x_1, \ldots, x_n\}$, kita partisi menjadi q *cluster*, misalkan, $\{C_1, \ldots, C_q\}$ dengan algoritma *cluster* seperti *k-means* atau *kernel k-means*. Sehingga, kita menggunakan $(x_i^l, y_i^l), i = 1, \ldots, n_l$ sebagai contoh indeks *cluster* ke – l, dimana $n_l, l = 1, \ldots, q$ merupakan banyaknya pengamatan pada *cluster* ke-l. Untuk setiap *cluster*, kita melakukan training dengan SVM, $f_l(x), 1 \leq l \leq q$. Model klasifikasi akhir didefinisikan sebagai fungsi indikator sebagai berikut

$$f(x) = \sum_{l=1}^{q} \boldsymbol{x}^T \boldsymbol{w}_l \, \mathbf{1} \, (\boldsymbol{x} \epsilon C_l). \tag{2.27}$$

dengan $\mathbf{1}(.)$ merupakan fungsi indikator. Pada kasus ini tidak menggunakan bias b. Sehingga formulasi *cluster* SVM

$$\min_{\boldsymbol{w}, \boldsymbol{w}_l, \xi_i^l \geq 0} \; \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2}\sum_{l=1}^{q}\|\boldsymbol{w}_l - \boldsymbol{w}\|^2 + C\sum_{l=1}^{q}\sum_{i=1}^{n_l}\xi_i^l, \tag{2.28}$$

fungsi kendala : $y_i^l \boldsymbol{w}_l^T x_i^l \geq 1 - \xi_i^l, i = 1, \ldots, n_l, \forall l,$

dengan ξ_i^l merupakan variabel *slack*, $\boldsymbol{w}$ merupakan vektor bobot referensi global, $\frac{1}{2}\sum_{l=1}^{q}\|\boldsymbol{w}_l - \boldsymbol{w}\|^2$ merupakan *global regularization*, dimana membutuhkan bobot linier lokal SVM ($\boldsymbol{w_l}$) sejalan dengan bobot referensi global. $\boldsymbol{w}$ menjembatani di antara kelompok yang berbeda, sehingga informasi dari satu *cluster* dapat dimanfaatkan ke yang lain. Oleh karena itu, hal tersebut dapat menghindari *over-fitting* di setiap *cluster* lokal. Secara khusus, jika kita menetapkan $\boldsymbol{w} = 0$, maka

CSVM (*Clustered Support Vector Machine*) akan menjadi q SVM independen yang dilatih di setiap *cluster* secara terpisah.

Diberikan $\boldsymbol{m}_l = \boldsymbol{w}_l - \boldsymbol{w}$, maka $\boldsymbol{w}_l = \boldsymbol{m}_l + \boldsymbol{w}$. Persamaan 2.28 dapat ditulis menjadi

$$\min_{\boldsymbol{w},\boldsymbol{m}_l,\xi_i^l \geq 0} \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2}\sum_{l=1}^{q}\|\boldsymbol{m}_l\|^2 + C\sum_{l=1}^{q}\sum_{i=1}^{n_l}\xi_i^l, \tag{2.29}$$

fungsi kendala : $y_i^l(\boldsymbol{m}_l + \boldsymbol{w})^T x_i^l \geq 1 - \xi_i^l, i = 1, \dots, n_l, \forall l.$

Untuk lebih menyederhanakan optimasi di atas masalah, kita definisikan $\widetilde{\boldsymbol{w}} = \left[\sqrt{\lambda}\boldsymbol{w}^T, \boldsymbol{m}_1^T, \dots, \boldsymbol{m}_q^T\right]^T$ dan $\widetilde{\boldsymbol{x}}_i^l = \left[\frac{1}{\sqrt{\lambda}}x_i^{lT}, \boldsymbol{0}^T \dots, x_i^{lT} \dots, \boldsymbol{0}^T\right]^T$ dimana setiap komponen ke $l+1$ dari $\widetilde{\boldsymbol{x}}_i^l$ adalah x_i^l. Sehingga permasalahan optimasi pada persamaan (2.29) dapat ditulis sebagai berikut

$$\min_{\boldsymbol{w},\boldsymbol{w}_l,\xi_i^l \geq 0} \frac{\lambda}{2}\|\widetilde{\boldsymbol{w}}\|^2 + C\sum_{l=1}^{q}\sum_{i=1}^{n_l}\xi_i^l, \tag{2.30}$$

fungsi kendala : $y_i^l \widetilde{\boldsymbol{w}}^T \widetilde{\boldsymbol{x}}_i^l \geq 1 - \xi_i^l, i = 1, \dots, n_l, \forall l.$

Sehingga untuk solusi *dual* pada permasalahan di atas dapat mengacu pada persamaan (2.20).

2.4 Analisis *Survival*

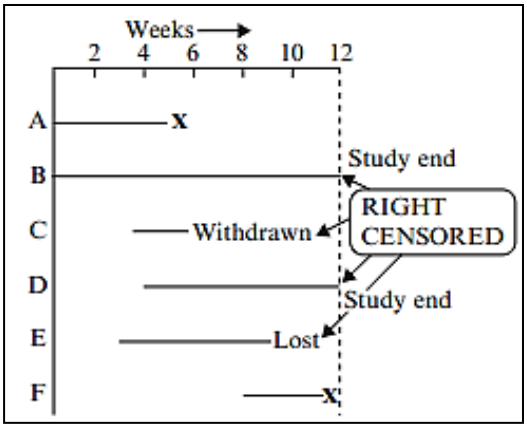
Analisis *survival* merupakan prosedur statistik untuk data analisis dengan *ouput* variabel berupa waktu hingga suatu kejadian (*event*) terjadi. Waktu kejadian dapat diukur dalam hari, minggu, bulan, tahun saat suatu pengamatan diamati hingga terjadinya suatu kejadian. Sedangkan *event* yang dimaksud adalah setiap pengalaman dari suatu observasi yang kemungkinan akan terjadi (Kleinbaum & Klein, 2012).

Karakteristik utama pada data survival adalah memiliki variabel respon yang *non negative* dan merepresentasikan waktu dari pengamatan mulai diamati hingga terjadinya suatu *event*. Kedua ialah data tersensor, yakni terjadi apabila peneliti tidak dapat mengetahui waktu *survival* pasti dari individu yang sedang diobservasi (Moore, 2016). Secara umum waktu yang menjadi fokus dalam analisis *survival* disebut survival time (T) dimana menunjukan waktu sebuah pengamatan

failure dalam periode tertentu. Sedangkan *event* dapat dianggap sebagai suatu kegagalan atau *failure* (δ) sebab kejadian yang biasanya diperhatikan adalah mengenai kematian, penyakit dan musibah lain yang dapat menimpa individu. Suatu *event* dilambangkan dengan simbol δ untuk mendefinisikan status *event* apakah *failure* atau tersensor. Nilai $\delta = 1$ menunjukan failure dan $\delta = 0$ menunjukan tersensor. Penyebab terdapat data tersensor secara umum disebabkan tiga hal yakni

a. Tidak ada *event* yang terjadi pada individu yang diobservasi hingga penelitian berakhir.
b. Selama periode observasi seseorang hilang dari pengamatan (*lost to follow up*).
c. Individu berhenti diobeservasi karena meninggal. Namun meninggalnya disebabkan hal lain yang tidak ada kaitannya dengan *event* yang diamati (*withdraws*).

Terdapat tiga jenis data tersensor yaitu data tersensor kanan, tersensor kiri dan tersensor interval. Namun kasus data tersensor yang sering terjadi ialah data tersensor kanan. Data tersensor kanan terjadi apabila tidak diketahui secara pasti *survival time* dari individu yang diamati dalam periode tertentu pengamatan. Penjelasan tentang data tersensor kanan dapat dijelaskan lebih mudah melalui gambar berikut



Gambar 2.5 Ilustrasi Kasus Data Tersensor Kanan

Gambar 2.5 menunjukan adanya data tersensor kanan saat dilakukan pengamatan terhadap 6 orang individu. Data pada individu B,C,D dan E tersensor

kanan disebabkan karena berakhirnya pengamatan, hilang dan *withdrawn* (Kleinbaum dan Klein, 2012).

Terdapat dua ukuran penting pada analisis *survival* yakni *survival function* yang dilambangkan dengan $S(t)$ dan *hazard function* dilambangkan dengan $h(t)$. *Survival function* merupakan probabilitas individu dapat bertahan lebih dari waktu tertentu, sedangkan *hazard function* adalah laju terjadinya *event* sesaat setelah individu bertahan hingga waktu tertentu. Secara matematis *survival function* dapat dinyatakan sebagai berikut.

$$S(t) = P(T > t) \tag{2.31}$$

Dengan T adalah waktu terjadinya event yang berupa variabel random kontinu maka *survival function* adalah komplemen dari fungsi distribusi kumulatif. Dimana fungsi distribusi kumulatif didefinisikan sebagai probabilitas variabel random *T* kurang dari atau sama dengan waktu *t* yang secara matematis dirumuskan $F(t) = P(T \leq t)$ sehingga *survival function* dapat dinyatakan sebagai berikut.

$$S(t) = P(T > t) = 1 - P(T \leq t) = 1 - F(t) \tag{2.32}$$

Jika dinyatakan dalam *probability density function* (PDF) *survival function* dapat dinyatakan sebagai berikut.

$$S(t) = P(T > t) = \int_{t}^{\infty} f(u)du \tag{2.33}$$

Ukuran penting yang kedua ialah *hazard function* didefinisikan sebagai *rate* suatu individu mengalami *event* pada interval waktu *t* hingga $t + \Delta t$ jika diketahui individu tersebut masih hidup sampai waktu *t.* Secara matematis *hazard function* dapat dirumuskan sebagai berikut.

$$h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T \geq t)}{\Delta t} \tag{2.34}$$

Hubungan antara *survival function* dan *hazard function* dapat menggunakan konsep probabilitas bersyarat $P(A|B) = \frac{P(A \cap B)}{P(B)}$, dimana A merupakan *hazard function* dan B merupakan *survival function*. Dan $P(A \cap B)$ adalah suatu probabilitas kejadian bersama antara A dan B. Nilai probabilitas bersyarat dari definisi fungsi *hazard* adalah sebagai berikut.

$$\frac{P(t \leq T < t + \Delta t)}{P(T > t)} = \frac{F(t + \Delta t) - F(t)}{S(t)} \tag{2.35}$$

dimana $F(t)$ adalah fungsi distribusi dari T, sehingga diperoleh,

$$h(t) = \lim_{\Delta t \to 0} \left\{ \frac{F(t+\Delta t) - F(t)}{\Delta t} \right\} \frac{1}{S(t)} \tag{2.36}$$

dengan,

$$F'(t) = f(t) = \lim_{\Delta t \to 0} \left\{ \frac{F(t + \Delta t) - F(t)}{\Delta t} \right\} \tag{2.37}$$

merupakan definisi turunan dari $F(t)$. Sehingga hubungan antara fungsi *survival* dan fungsi *hazard* adalah sebagai berikut.

$$h(t) = \frac{f(t)}{S(t)} \tag{2.38}$$

Jika $F(t) = 1 - S(t)$ maka $f(t) = \frac{d(F(t))}{dt} = \frac{d(1-S(t))}{dt}$ sehingga nilai $h(t)$ dapat dinyatakan sebagai berikut.

$$\begin{aligned} h(t) &= \frac{\left(\frac{d(1 - S(t))}{dt}\right)}{S(t)} \\ &= \frac{\left(\frac{-d(S(t))}{dt}\right)}{S(t)} \\ &= -\frac{d\big(S(t)\big)}{dt} \cdot \frac{d\ ln\big(S(t)\big)}{dS(t)} \\ -h(t) &= \frac{d\ ln\big(S(t)\big)}{dt} \end{aligned} \tag{2.39}$$

Sehingga jika kedua ruas fungsi diintegralkan akan diperoleh hubungan antara fungsi *h(t)* dan fungsi *S(t)* sebagai berikut.

$$\begin{aligned} -\int_0^t h(u)du &= \int_0^t \frac{1}{S(u)} d\left(S(u)\right) \\ &= \ln S(u) \Big|_0^t \\ &= \ln S(t) - \ln S(0) \\ &= \ln S(t) \end{aligned} \tag{2.40}$$

Hubungan antara *hazard function* dan *survival function* dapat dirumuskan sebagai berikut.

$$H(t) = -\ln S(t) \tag{2.41}$$

Sehingga, fungsi *survival* dapat dituliskan sebagai berikut.

$$S(t) = \exp\big(-H(t)\big) \tag{2.42}$$

Dengan $H(t)=\int_0^t h(u)du$

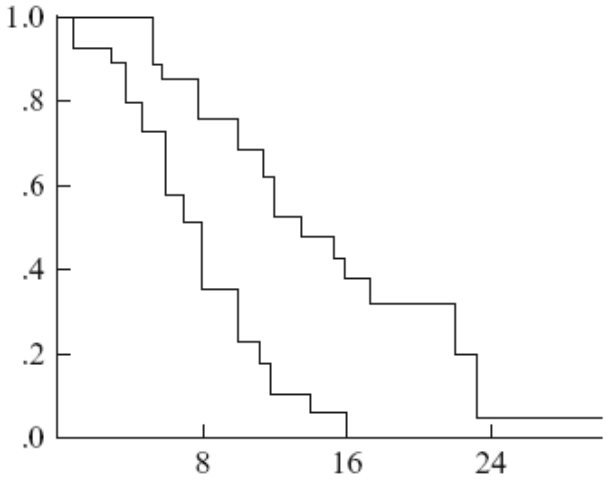
2.5 Kurva *Kaplan Meier* dan Uji *Log Rank*

Kurva Kaplan Meier digunakan untuk mengestimasi dan menggambarkan kurva fungsi *survival* yang menghubungkan antara estimasi fungsi *survival* dengan waktu *survival* (Kleinbaum dan Klein, 2012). Apabila probabilitas dari Kaplan Meier adalah $\hat{S}(t_{(j)})$ maka persamaan umumnya adalah sebagai berikut.

$$\widehat{S}(\mathrm{t}_{(j)}) = \widehat{S}(\mathrm{t}_{(j-1)})\widehat{P}(T > t_{(j)} \mid T \geq t_{(j)}) \tag{2.43}$$

$$\widehat{S}(\mathrm{t}_{(j-1)}) = \prod_{i=1}^{j-1} \widehat{P}(T > t_{(j)} \mid T \geq t_{(j)}) \tag{2.44}$$

Ilustrasi kurva survival Kaplan Meier dapat dilihat pada Gambar 2.6 berikut.



Gambar 2.6 Ilustrasi Kurva *Survival* Kaplan Meier (Kleinbaum, D. G., dan Klein, M. 2012)

Uji Log Rank merupakan uji yang digunakan untuk membandingkan kurva *survival* dalam grup yang berbeda (Kleinbaum dan Klein, 2012).

Hipotesis dari uji Log Rank untuk dua grup atau lebih adalah sebagai berikut.

H_0 : tidak ada perbedaan kurva *survival* dalam grup yang berbeda

H_1 : paling sedikit ada satu perbedaan kurva *survival* dalam grup yang berbeda

Statistik Uji :

$$\chi^2 = \sum_{h=1}^{G} \frac{(O_h - E_h)^2}{E_h} \tag{2.45}$$

Dimana

$$O_h - E_h = \sum_{j=1}^{G} (m_{hj} - e_{hj}) \text{ dan } e_{hj} = \left(\frac{n_{hj}}{\sum_{h=1}^{G} n_{hj}} \right) \left(\sum_{h=1}^{G} m_{hj} \right)$$

m_{hj} = jumlah subjek yang gagal dalam grup ke-*h* pada waktu $t_{(j)}$

n_{hj} = jumlah subjek yang beresiko gagal seketika pada grup ke-*h* sebelum waktu $t_{(j)}$

e_{hj} = nilai ekspektasi dalam grup ke-*h* pada waktu $t_{(j)}$

G = banyak grup

Tolak H_0 jika nilai $\chi^2 > \chi^2_{\alpha,G-1}$

2.6 Model Regresi *Cox Proportional Hazard*

Model Regresi *Cox Proportional Hazard* pertama kali dikenalkan oleh D.R. Cox (1972). Model ini merupakan model semiparametric karena *baseline hazard* yang dinyatakan sebagai $h_0(t)$ tidak mengharuskan mengikuti distribusi tertentu. Persamaan regresi *Cox Proportional Hazard* sebagai berikut

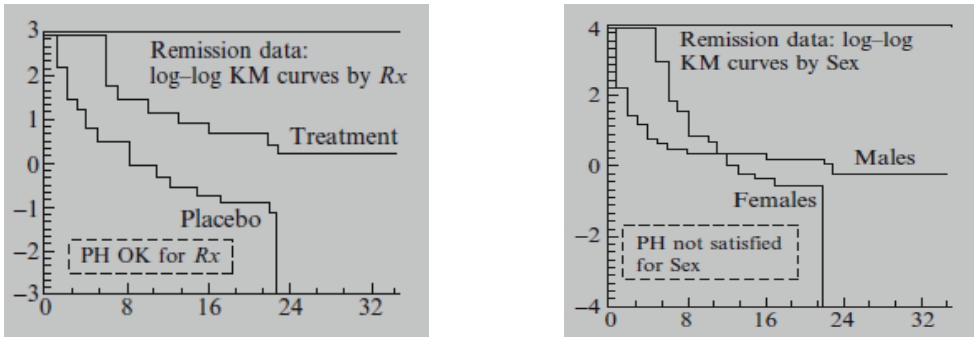
$$h(t, \boldsymbol{x}) = h_0(t) \exp(\boldsymbol{\beta}^T \boldsymbol{x}) \tag{2.46}$$

dengan $\boldsymbol{x}$ merupakan vektor variabel prediktor $(x_1, x_2, \ldots, x_k)^T$ sedangkan $\boldsymbol{\beta}$ merupakan koefisien parameter untuk variabel prediktor $\boldsymbol{x}$ (Kleinbaum dan Klein, 2012).

Asumsi yang penting pada regresi *cox proportional hazard* adalah memiliki proporsi *hazard function* yang konstan untuk setiap waktu. Untuk mengetahui asumsi terpenuhi terdapat dua acara yakni secara grafis dan menggunakan pengujian *goodness of fit* (Machin et. al, 2006).

a. Grafik

Secara grafik, asumsi *cox proportional hazard* dapat dilihat dengan cara membandingkan antara $\ln[-\ln(S(t))]$ atau $-\ln[-\ln(S(t))]$ untuk masing-masing variabel prediktor kategorik. Sebagai ilustrasi dapat dilihat pada Gambar 2.7.



(a) (b)

Gambar 2.7 Ilustrasi Asumsi *Cox Proportional Hazard* secara Grafik. (a) Memenuhi Asumsi, (b) Tidak Memenuhi Asumsi (Kleinbaum dan Klein, 2012).

Berdasarkan Gambar 2.7 (a), terlihat bahwa asumsi *proportional hazard* terpenuhi karena garis yang mewakili data *placebo* cenderung sejajar dengan garis yang mewakili data *treatment* (Kleinbaum dan Klein, 2012).

b. Uji *Goodness of Fit*

Pengujian asumsi *cox proportional hazard* dapat menggunakan Residual Schoenfeld. Tahapan pengujian asumsi *proportional hazard* menggunakan residual *Schoenfeld* adalah sebagai berikut (Schoenfeld, 1982).

1. Membangun model Cox *proportional hazard* dengan metode *goodness of fit* menggunakan residual *Schoenfeld* untuk setiap variabel prediktor.
2. Membuat variabel rank *survival time* dimana waktu *survival* diurutkan mulai dari individu yang mengalami *event* pertama kali.
3. Menguji korelasi antara variabel yang dihasilkan pada langkah pertama yaitu residual *Schoenfeld* dengan variabel yang dihasilkan pada langkah kedua yaitu *rank survival time* (Kleinbaum dan Klein, 2012).

Residual *Schoenfeld* dari variabel prediktor ke-*k* dan individu yang mengalami *event* pada waktu $t_{(j)}$ didefinisikan sebagai berikut.

$$SCH_{kj} = x_{kj} - E(x_{kj} \mid R(t_{(kj)})) \tag{2.47}$$

Dimana

$$E(x_{kj} \mid R(t_{(kj)})) = \frac{\sum_{l \in R(t_{(j)})} x_{kj} \exp(\boldsymbol{\beta}^T \boldsymbol{x}_l)}{\sum_{l \in R(t_{(j)})} \exp(\boldsymbol{\beta}^T \boldsymbol{x}_l)} \tag{2.48}$$

SCH_{kj} : residual *Schoenfeld* untuk variabel prediktorke-*k* individu yang mengalami *event* pada waktu $t_{(j)}$

x_{kj} : nilai dari variabel prediktor ke-*k* dari individu yang mengalami *event* pada waktu $t_{(j)}$

$E\left(x_{kj} \middle| R\left(t_{(kj)}\right)\right)$: *conditional expectation* x_{kj} jika $R_{t(j)}$ diketahui,

$E\left(x_{pj} \middle| R\left(t_{(pj)}\right)\right)$: *conditional expectation* x_{pj} jika $R_{t(j)}$ diketahui,

Dalam pengujian korelasi antara residual *Schoenfeld* dengan *rank survival time* masing-masing variabel prediktor digunakan koefisien korelasi pearson.

$$\rho = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^{\,2} - \left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{n\sum_{i=1}^{n} y_i^{\,2} - \left(\sum_{i=1}^{n} y_i\right)^2}} \tag{2.49}$$

dengan

n : banyaknya individu

x_i : residual *Schoenfeld* individu ke-*i*

y_i : *rank survival time* individu ke-*i*

Statistik uji korelasi pearson

$H_0 : \rho = 0$

$H_1 : \rho \neq 0$

$$t = \frac{r}{\sqrt{\frac{1-r^2}{n-2}}} \tag{2.50}$$

Tolak H_0, jika $|t| > t_{(\alpha/2,n-1)}$

c. Variabel *Time Dependent*

Pemeriksaan asumsi *proportional hazard* juga dapat dilakukan dengan uji variabel *time dependent.* Variabel *time dependent* adalah variabel prediktor dalam model Cox *proportional hazard* yang diinteraksikan dengan fungsi waktu.

2.7 Survival SVM

Metodologi menggunakan SVM untuk analisis data *survival*, pertama kali diteliti oleh (Van Belle, *et al.*, 2007 & 2008). Sebagai ganti dari fungsi *hazard*, kesesuaian antara observasi waktu kegagalan dan *output* model dioptimalkan. *Output* model merupakan fungsi prognostik juga disebut sebagai fungsi utilitas dan lebih spesifik dalam penelitian medis disebut prognostik indeks atau fungsi kesehatan dimana $u: \mathbb{R}^D \rightarrow \mathbb{R}$ didefinisikan sebagai berikut

$$u(\boldsymbol{x}) = \boldsymbol{w}^T \varphi(\boldsymbol{x}), \tag{2.51}$$

di mana *w* vektor parameter yang tidak diketahui dan $\varphi(x)$ merupakan transformasi dari kovariat x.

Concordance Index (C-index) yang diperkenalkan oleh Harrell (1984) digunakan untuk mengukur *Concordance* antara fungsi utilitas dan waktu kegagalan observasi menggunakan observasi tersensor dan tidak tersensor. *Concordance* yang empiris antara waktu terjadinya *event* (t_i) dan prognostik index u_i berdasarkan dataset $D = \{(x_i, t_i, \delta_i)\}_{i=1}^n$ didefinisikan sebagai berikut.

$$c_{ij}(u) = \frac{\sum_{i=1}^{n}\sum_{j>i}^{n} v_{ij} I\left(\left(u(x_j) - u(x_i)\right)(t_j - t_i) > 0\right)}{\sum_{i=1}^{n}\sum_{j>i}^{n} v_{ij}}, \tag{2.52}$$

dimana $\left(\left(u(x_j) - u(x_i)\right)(t_j - t_i) > 0\right)$merupakan fungsi indikator dan v_{ij} dirumuskan sebagai berikut.

$$v_{ij} = \begin{cases} 1, \ (t_i < t_j \text{ dan } \delta_i = 1) \text{ atau } (t_j < t_i \text{ dan } \delta_j = 1) \\ 0, \ \text{untuk yang lainnya} \end{cases}$$

dengan (x_i, t_i, δ_i) dan $\left(x_j, t_j, \delta_j\right)$akan dibandingkan ketika peringkat dalam domain waktu diketahui.

Diasumsikan bahwa $t_i < t_j$ untuk $i < j$. Jika asumsi tersebut dilanggar, maka data diurutkan terlebih dahulu. Dengan mengetahui bahwa t_i lebih kecil dari t_j, diharapkan bahwa nilai $u(x_i)$ juga lebih kecil dari $u(x_j)$. Sehingga

$$u(x_j) - u(x_i) \geq 1, \forall i < j \tag{2.53}$$

Model *survival* SVM diimplikasikan dengan fungsi kendala (*constraint*) yang akan mendapatkan margin yang tepat. Fungsi kendala model *survival* SVM ditunjukkan pada persamaan (2.53) di atas. Jika terjadi kesalahan dalam memberi peringkat maka diberi variabel slack yaitu $\xi_{ij} \geq 0$. Sehingga formulasi model *survival* SVM adalah sebagai berikut

$$\min_{w,\xi} \ \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{i<j} v_{ij}\xi_{ij}\,, \tag{2.54}$$

$$\text{fungsi kendala } \begin{cases} \boldsymbol{w}^T\varphi(x_j) - \boldsymbol{w}^T\varphi(x_i) \geq 1 - \xi_{ij}, \forall\ i < j \\ \xi_{ij} \geq 0,\ \forall\ i < j \end{cases}, \tag{2.55}$$

dengan $\gamma \geq 0$ dan w didefinisikan pada persamaan (2.51).

Van belle (2011) kemudian mengembangkan dengan menggabungkan batasan berbasis *ranking* dari *concordance index* dan pendekatan regresi. Ide ini berawal dari persamaan regresi merupakan kombinasi linier antara variabel *output* y dengan kovariat

$$y = \boldsymbol{w}^T x + \epsilon\,,1 \tag{2.56}$$

dimana ϵ merupakan variabel *error*. Jika variabel y tidak mengikuti distribusi normal, maka persamaan regresi dapat dilakukan dengan melakukan transformasi pada variabel y. Kemudian jika terdapat hubungan *non* linier variabel x dapat ditransformasi sehingga menjadi

$$h(y) = \boldsymbol{w}^T\varphi(x) + \epsilon. \tag{2.57}$$

Pada model survival berbasis distribusi seperti Weibull, *error* merupakan selisih antara *hazard function* dengan prognostik indeks

$$\epsilon = h(y) - \boldsymbol{w}^T\varphi(x). \tag{2.58}$$

Berdasarkan kondisi tersebut, formulasi model survival SVM berbasis *ranking* dari *concordance index* dan regresi menjadi

$$\min_{w,\epsilon,\xi,\xi^*,b} \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\sum_{i=1}^{n}\epsilon_i + \mu\sum_{i=1}^{n}(\xi_i + \xi_i^*), \tag{2.59}$$

$$\text{fungsi kendala } \begin{cases} w^T(\varphi(x_i) - \varphi(x_{j(i)})) \geq y_i - y_{j(i)} - \varepsilon_i, \\ w^T\varphi(x_i) + b \geq y_i - \xi_i, \\ -\delta_i(w^T\varphi(x_i) + b) \geq \delta_i y_i - \xi_i^*, \\ \varepsilon_i \geq 0 \\ \xi_i \geq 0 \\ \xi_i^* \geq 0 \end{cases} \quad \forall i = 1,\ldots,n. \tag{2.60}$$

Kemudian fungsi *lagrangian* menjadi

$$\begin{aligned} \mathcal{L}(w,\epsilon,b,\alpha,b) = {} & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \gamma\sum_{i=1}^{n}\epsilon_i + \mu\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ & - \sum_{i=1}^{n}\alpha_i\left(\boldsymbol{w}^T\left(\varphi(x_i) - \varphi\left(x_{j(i)}\right)\right) - y_i + y_{j(i)} + \epsilon_i\right) \\ & - \sum_{i=1}^{n}\beta_i(\boldsymbol{w}^T\varphi(x_i) + b - y_i + \xi_i) \\ & - \sum_{i=1}^{n}\beta_i^*\left(-\delta_i\left(\boldsymbol{w}^T\varphi(x_i)\right) + b - \delta_i y_i + \xi_i^*\right) \\ & - \sum_{i=1}^{n}\eta_i\epsilon_i - \sum_{i=1}^{n}\nu_i\xi_i - \sum_{i=1}^{n}\nu_i^*\xi_i^* \end{aligned} \tag{2.61}$$

Setelah dilakukan optimasi menggunakan Karush-Tuhn-Tucker (KKT) diperoleh persamaan prognostik indeks $u(\boldsymbol{x}^*)$

$$\hat{u}(x^*) = \sum_{i=1}^{n}(\alpha_i\left(\varphi(x_i) - \varphi\left(x_{j(i)}\right)\right) + (\beta_i - \delta_i\beta_i^*)\varphi(x_i))^T\varphi_p(x^*) + b \tag{2.62}$$

2.8 Kriteria Kebaikan Model

Terdapat dua tipe kriteria kebaikan model, yakni untuk tahap klasifikasi dan tahap analisis *survival*. Evaluasi kebaikan model pada tahap klasifikasi adalah *ROC* (*Receiver Operating Characteristic*) *Curve* dan AUC (*Area Under Curve*). *ROC* dan AUC dapat ditentukan menggunakan nilai yang terdapat dalam *confusion matrix*. *Confusion matrix* adalah tabulasi silang antara aktual dan prediksi (Giudici, 2003). *Confusion matrix* ditunjukkan pada Tabel 2.1.

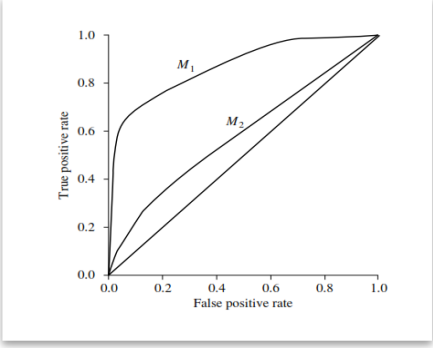
Tabel 2.1.Confussion Matrix

Aktual	Prediksi	
	Event (1)	Non-Event (0)
Event (1)	*True Positives* (TP)	*False Negatives* (FN)
Non-Event (0)	*False Positives* (FP)	*True Negatives* (TN)

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

$$Specificity = \frac{TN}{FP + TN}$$

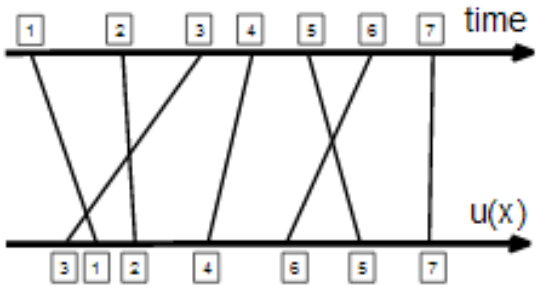
ROC adalah sebuah alat visual yang berguna untuk membandingkan model klasifikasi. Suatu kurva ROC untuk suatu model *classifier* menunjukan *trade-off* diantara *true positive rate* (TPR) dan *false positive rate* (FPR). TPR merupakan *sensitivity* sedangkan FPR merupakan 1-*specificity*. ROC memanfaatkan informasi probabilitas dari hasil prediksi pengamatan ke suatu kelas yang diurutkan dari yang terbesar ke terkecil untuk kemudian melakukan perhitungan TPR dan FPR pada masing masing pengamatan.



Gambar 2.8 Ilustrasi Kurva ROC untuk Dua Model Klasifikasi, M_1 dan M_2 (Han, Jiawei, dkk, 2012)

Berdasarkan ilustrasi pada Gambar 2.8 model klasifikasi M_1 lebih baik dibanding model M_2. Model klasifikasi yang paling dekat dengan garis diagonal pada kurva adalah model yang kurang akurat. Untuk menaksir akurasi dari model, dapat menggunakan ukuran luas area di bawah kurva ROC yaitu AUC (*Area Under Curve*) (Han dkk, 2012).

Kriteria kebaikan model untuk tahap analisis survival yakni *concordance index*. Pada perumusan C-indeks diasumsikan bahwa $t_i < t_j$ untuk $i < j$. Jika asumsi tersebut dilanggar, maka data diurutkan terlebih dahulu. Dengan mengetahui bahwa t_i lebih kecil dari t_j, diharapkan bahwa nilai $u(x_i)$ juga lebih kecil dari $u(x_j)$. Contoh sederhana cara perhitungan c-indeks secara manual.



Gambar 2.9 Ilustrasi Perhitungan C-Indeks (Van Bell *et al.*, 2011)

Misalkan kasus yang terjadi pada Gambar 2.8, dimana terdapat 7 pasien yang memiliki *survival time* dan prognostik indeks yang telah diperingkat. Berdasarkan Gambar 2.8 dapat diketahui bahwa perhitungan C-indeks adalah sebagai berikut.

Tabel 2.2 Ilustrasi Perhitungan C-Indeks

Concordance	*Ranking* 1 2 3 4 5 6	Jumlah
2	1	1
3	0 0	0
4	1 1 1	3
5	1 1 1 1	4
6	1 1 1 1 0	4
7	1 1 1 1 1 1	6
Jumlah	5 4 4 3 1 1	c-indeks 18/21

Sumber : Van Bell *et al.* (2011)

Berdasarkan ilustrasi pada Tabel 2.1, terdapat 3 *misranking*. Sebagai contoh peringkat dari prognostik indeks yang *survival time*nya peringkat 3 adalah 1 atau

yang memiliki prognostik indeks paling kecil. Itu artinya t_i lebih kecil dari t_j, diharapkan bahwa nilai $u(x_i)$ juga lebih kecil dari $u(x_j)$ dilanggar (*misranking*). Dari 21 kemungkinan pasangan peringkat $(t_i, u(x_i))$ dan $(t_j, u(x_j))$ ada 18 yang tidak *misranking*. Sehingga nilai C-indeks pada ilustrasi di atas adalah $\frac{18}{21}$. Semakin tinggi nilai C-indeks maka performansi dari metode yang digunakan semakin bagus (Mahjub, Hossein *et al*., 2016).

BAB 3

METODOLOGI PENELITIAN

3.1 Sumber Data

Sumber data yang digunakan pada penelitian ini terbagi menjadi dua yakni untuk simulasi dan data riil. Data simulasi bertujuan untuk menunjukkan bagaimana performansi metode klasifikasi *Ensemble* SVM dengan metode SVM. Data simulasi yang digunakan terdiri dari variabel respon (Y) yang berupa kategori klasifikasi, dan variabel prediktor (X). Berikut skenario karakteristik variabel dan nilai parameter yang akan dibangkitkan menjadi data simulasi.

Tabel 3.1 Skenario Data Simulasi

Skenario	Distribusi	Parameter	
1	$X_1,\ldots,X_4 \sim BIN(1, 0.2)$ $X_5 \sim N(20, 5)$ $X_6 \sim N(30, 5)$	$\beta_1 = 2$ $\beta_2 = 4$ $\beta_3 = 3$	$\beta_4 = 0$ $\beta_5 = 7$ $\beta_6 = -4$
2	$X_1,\ldots,X_4 \sim BIN(1, 0.2)$ $X_{5\_1} \sim N(20, 5)$ $X_{5\_2} \sim N(50, 5)$ $X_{6\_1} \sim N(10, 5)$ $X_{6\_2} \sim N(30, 5)$	$\beta_1 = 2$ $\beta_2 = 4$ $\beta_3 = 3$	$\beta_4 = 0$ $\beta_5 = 4$ $\beta_6 = -9$
3	$X_1,\ldots,X_4 \sim BIN(1, 0.2)$ $X_{5\_1} \sim N(20, 5)$ $X_{5\_2} \sim N(50, 5)$ $X_{6\_1} \sim N(10, 5)$ $X_{6\_2} \sim N(30, 5)$ $X_{7\_1}, X_{8\_1} \sim \boldsymbol{N}\left(\begin{pmatrix}20\\40\end{pmatrix}, \begin{pmatrix}25 & 16\\16 & 25\end{pmatrix}\right)$ $X_{7\_2}, X_{8\_2} \sim \boldsymbol{N}\left(\begin{pmatrix}50\\70\end{pmatrix}, \begin{pmatrix}25 & 16\\16 & 25\end{pmatrix}\right)$	$\beta_1 = 2$ $\beta_2 = 4$ $\beta_3 = 3$ $\beta_4 = 0$	$\beta_5 = 4$ $\beta_6 = -4$ $\beta_7 = 8$ $\beta_8 = -6$
4	$X_1,\ldots,X_4 \sim BIN(1, 0.2)$ $X_{5\_1} \sim N(20, 5)$ $X_{5\_2} \sim N(50, 5)$ $X_{6\_1} \sim N(10, 5)$ $X_{6\_2} \sim N(30, 5)$ $X_7 = 5X_5 - 3X_6$	$\beta_1 = 2$ $\beta_2 = 4$ $\beta_3 = 3$	$\beta_4 = 0$ $\beta_5 = 13$ $\beta_6 = -3$ $\beta_7 = -3.5$

Tabel 3.1 Skenario Data Simulasi (lanjutan)

Skenario	Distribusi	Parameter	
5	$X_1, \dots, X_4 \sim BIN(1, 0.2)$ $X_{5\_1} \sim N(20, 5)$ $X_{5\_2} \sim N(50, 5)$ $X_{6\_1} \sim N(10, 5)$ $X_{6\_2} \sim N(30, 5)$ $X_7 = 5X_5 - 3X_6$ $X_{8\_1}, X_{9\_1} \sim \boldsymbol{N}\left(\begin{pmatrix}20\\40\end{pmatrix}, \begin{pmatrix}25 & 16\\16 & 25\end{pmatrix}\right)$ $X_{8\_2}, X_{9\_2} \sim \boldsymbol{N}\left(\begin{pmatrix}10\\10\end{pmatrix}, \begin{pmatrix}25 & 16\\16 & 25\end{pmatrix}\right)$	$\beta_1 = 2$ $\beta_2 = 4$ $\beta_3 = 3$ $\beta_4 = 0$	$\beta_5 = 13$ $\beta_6 = -3$ $\beta_7 = -3.5$ $\beta_8 = 5.5$ $\beta_5 = -3$
6	$X_1 \sim N(20, 5)$ $X_2 \sim N(30, 5)$	$\beta_1 = 6$ $\beta_2 = -4$	
7	$X_{1\_1} \sim N(20, 5)$ $X_{1\_2} \sim N(50, 5)$ $X_{2\_1} \sim N(10, 5)$ $X_{2\_2} \sim N(30, 5)$	$\beta_1 = 3$ $\beta_2 = -5$	
8	$X_{1\_1} \sim N(20, 5)$ $X_{1\_2} \sim N(50, 5)$ $X_{2\_1} \sim N(10, 5)$ $X_{2\_2} \sim N(30, 5)$ $X_{3\_1}, X_{4\_1} \sim \boldsymbol{N}\left(\begin{pmatrix}20\\40\end{pmatrix}, \begin{pmatrix}25 & 16\\16 & 25\end{pmatrix}\right)$ $X_{3\_2}, X_{4\_2} \sim \boldsymbol{N}\left(\begin{pmatrix}50\\70\end{pmatrix}, \begin{pmatrix}25 & 16\\16 & 25\end{pmatrix}\right)$	$\beta_1 = 4$ $\beta_2 = -5$	$\beta_3 = -6$ $\beta_4 = 3$
9	$X_{1\_1} \sim N(20, 5)$ $X_{1\_2} \sim N(50, 5)$ $X_{2\_1} \sim N(10, 5)$ $X_{2\_2} \sim N(30, 5)$ $X_3 = 5X_1 - 3X_2$	$\beta_1 = 4$ $\beta_2 = 4.5$ $\beta_3 = -2$	
10	$X_{1\_1} \sim N(20, 5)$ $X_{1\_2} \sim N(50, 5)$ $X_{2\_1} \sim N(10, 5)$ $X_{2\_2} \sim N(30, 5)$ $X_3 = 5X_1 - 3X_2$ $X_{4\_1}, X_{5\_1} \sim \boldsymbol{N}\left(\begin{pmatrix}20\\40\end{pmatrix}, \begin{pmatrix}25 & 16\\16 & 25\end{pmatrix}\right)$ $X_{4\_2}, X_{5\_2} \sim \boldsymbol{N}\left(\begin{pmatrix}10\\10\end{pmatrix}, \begin{pmatrix}25 & 16\\16 & 25\end{pmatrix}\right)$	$\beta_1 = 6$ $\beta_2 = 6$	$\beta_3 = -3$ $\beta_4 = 5$ $\beta_5 = -2$

Berdasarkan Tabel 3.1 skenario 1 sampai 5, variabel prediktor berupa data kategori dan variabel kontinu. Sedangkan pada skenario 6 sampai 10, variabel prediktor yang dibangkitkan hanya variabel kontinu. Terdapat berbagai macam kondisi skenario variabel prediktor kontinu yang dibangkitkan yakni kondisi variabel prediktor berdisitribusi normal univariat, *mixture* normal univariat, *mixture* normal multivariat, dan kombinasi linier dari *mixture* normal univariat. Cara membangkitkan data *mixture* yakni dengan masing-masing membangkitan separuh data ($n_1, n_2 = 250$) memiliki parameter lokasi yang berbeda. Hal tersebut ditunjukkan dengan variabel yang memiliki indeks seperti pada skenario 2 (X_{2_1} dan X_{2_2}).

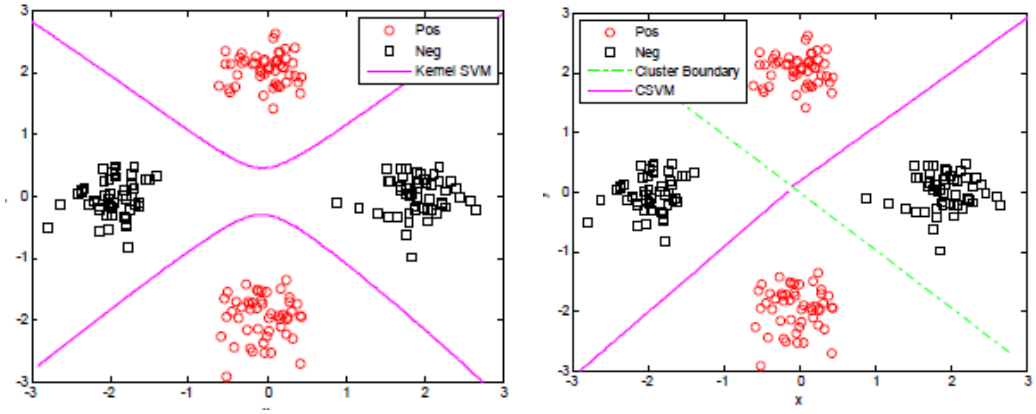
Variabel respon dibangkitkan berdasarkan persamaan (2.7) sehingga variabel respon mengikuti distribusi Bernoulli dengan 500 pengamatan (observasi). Setiap skenario, proses membangkitkan data diulang sebanyak 100 kali. Sehingga total dataset yang digunakan pada kasus data simulasi yaitu 1000 dataset dengan masing-masing 500 observasi pada tiap dataset.

Data riil diperoleh dari perusahaan *fintech* X. Data tersebut merupakan data nasabah gadai yang melakukan transaksi dari bulan April tahun 2015 hingga Agustus 2017. Adapun variabel yang diperoleh adalah durasi nasabah melakukan pembayaran lebih cepat / terlambat membayar, usia nasabah (X_1), besaran peminjaman (X_2), durasi peminjaman (X_3), jenis kelamin nasabah (X_4), nasabah baru / lama (X_5), nasabah masih memiliki tanggungan pinjaman atau tidak (X_6), durasi *early / late payment* (T), dan kategori nasabah (Y). Total banyaknya transaksi pada data tersebut adalah 1759 transaksi.

3.2 Ilustrasi Metode *Ensemble* SVM

Metode *Ensemble* SVM yang digunakan pada penelitian kali ini berbasis klaster pada observasi. Perbedaan utama antara *Ensemble* SVM dan SVM pada umumnya adalah proses untuk memecahkan solusi *non linier*. Pada SVM solusi pemecahan non linier dapat diselesaikan dengan *kernel trick*. Salah satu metode kernel yang umum digunakan adalah *kernel* RBF. Namun *kernel* SVM memiliki komputasi yang cukup besar. Sehingga apabila dimensi dari suatu data semakin besar, maka proses *training* semakin lama. Oleh sebab itu Gu dan Han (2013)

memperkenalkan *Ensembe* SVM berbasis klaster pada observasi. Sehingga permasalahan data yang non linier dapat diselesaikan secara linier tanpa transformasi menggunakan *kernel*. Metode klaster yang dapat digunakan bermacam-macam. Beberapa metode yang umum digunakan adalah *k-means* dan *kernel k-means.* Tahapan dalam melakukan Ensemble SVM ini adalah (i) memperoleh klaster untuk masing-masing data pengamatan, (ii) mendapatkan matrix transformasi $\tilde{\boldsymbol{x}}$, (iii) mendapatkan nilai $\breve{\boldsymbol{w}}$ dengan mengacu persamaan (2.29), (iv) mendapatkan hasil klasifikasi. Ilustrasi penyelesaian kasus non linier pada kernel SVM dan *Ensemble* SVM dapat dilihat pada Gambar 3.1



(a) (b)

Gambar 3.1 Ilustrasi Kernel SVM (a) dan Ensemble SVM (b) pada XOR Dataset (Gu dan Han, 2013)

3.3 Struktur Data

Struktur data terbagi menjadi dua tipe yakni struktur untuk tahap klasifikasi dan tahap analisis *survival*. Sebagai ilustrasi Tabel 3.2 merupakan struktur data untuk tahap klasifikasi data simulasi. Tabel 3.3 merupakan struktur data untuk tahap klasifikasi data riil. sedangkan Tabel 3.4 merupakan struktur data untuk analisis *survival* untuk nasabah *early payment* maupun *late payment*. Pada analisis survival, status nasabah tidak ada data tersensor sehingga nilai δ untuk masing-masing nasabah adalah 1.

Tabel 3.2 Struktur Data Studi Simulasi

Pengamatan ke	**Y**	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$	**…**	$\mathbf{X_k}$
1	Y_1	X_{11}	X_{12}	X_{13}	X_{14}	…	X_{1k}
2	Y_2	X_{21}	X_{22}	X_{23}	X_{24}	…	X_{2k}
3	Y_3	X_{31}	X_{32}	X_{33}	X_{34}	…	X_{3k}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	Y_n	X_{n1}	X_{n2}	X_{n3}	X_{n4}	…	X_{nk}

Tabel 3.3 Struktur Data Riil Tahap klasifikasi

Transaksi ke	**Y (*late payment*=1, *early payment*/tepat waktu=0)**	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$	$\mathbf{X_5}$	$\mathbf{X_6}$
1	Y_1	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}
2	Y_2	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}
3	Y_3	X_{31}	X_{32}	X_{33}	X_{34}	X_{35}	X_{36}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	Y_n	X_{n1}	X_{n2}	X_{n3}	X_{n4}	X_{n5}	X_{n6}

Tabel 3.4 Struktur Data Tahap Analisis *Survival* untuk Nasabah *Early Payment* / *Late Payment*

Transaksi ke	***Survival Time* (hari)**	$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{X_3}$	$\mathbf{X_4}$	$\mathbf{X_5}$	$\mathbf{X_6}$
1	t_1	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}
2	t_2	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}
3	t_3	X_{31}	X_{32}	X_{33}	X_{34}	X_{35}	X_{36}
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
N	t_n	X_{n1}	X_{n2}	X_{n3}	X_{n4}	X_{n5}	X_{n6}

Keterangan variabel data riil:

X_1 (Tahun) X_4 (Laki-laki=1, Perempuan=0)

X_2 (Rp dalam ribuan) X_5 (Nasabah baru=1, lama=0)

X_3 (dalam minggu) X_6 (Memiliki pinjaman sebelumnya =1, tidak memiliki=0)

3.4 Tahapan Penelitian

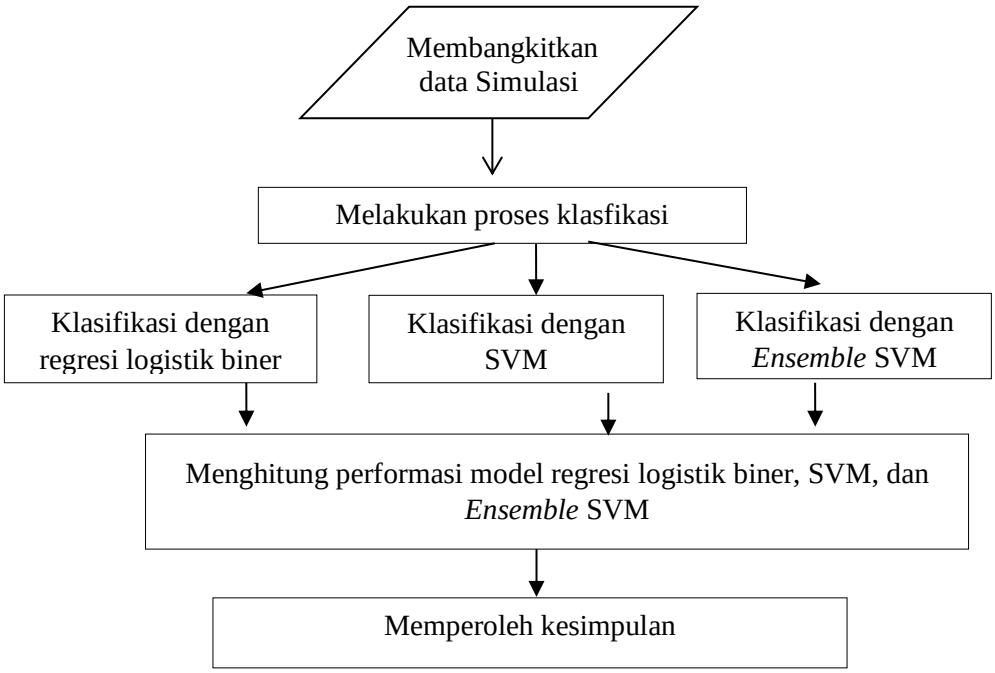
Tahapan penelitian ini terbagi menjadi tiga macam yakni menjelaskan langkah-langkah *Ensemble* SVM, studi simulasi untuk kasus klasifikasi dan kasus data riil.

a. Menjelaskan Langkah-langkah *Ensemble* SVM
   1. Membangkitkan data ilutstrasi.
   2. Melakukan klaster pada data
   3. Melakukan transformasi pada variabel $\tilde{\boldsymbol{x}}$
   4. Mendapatkan vektor $\tilde{\boldsymbol{w}}$ dan memperoleh hasil klasifikasi

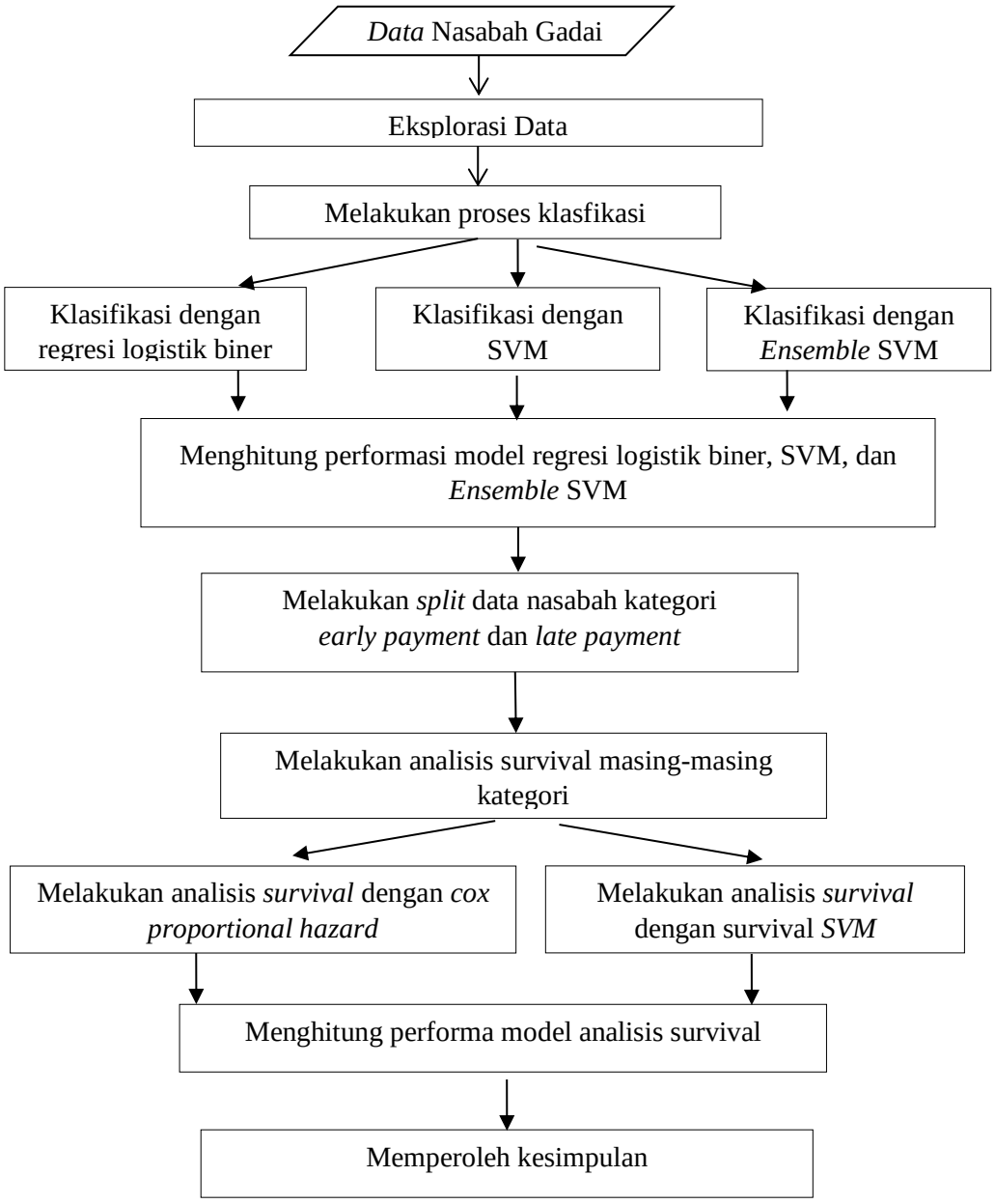
b. Tahap Studi Simulasi
   5. Membangkitkan data sesuai dengan skenario pada Bab 3.1
   6. Melakukan proses klasifikasi dengan Regresi Logistik, SVM dan *Ensemble SVM* dengan algoritma *cluster kmeans* dan *kernel kmeans* dimana banyaknya *cluster* ditentukan sebanyak 2, 3 dan 4.
   7. Mengevaluasi model Regresi Logistik, SVM, dan *Ensemble* SVM dengan menggunakan akurasi, sensitivity, specificity, dan AUC.

c. Tahap Data Riil
   1. Melakukan analisis eksplorasi data.
   2. Melakukan proses klasifikasi dengan regresi logistik biner, SVM, dan *Ensemble* SVM. Banyak klaster yang dibentuk pada *Ensemble* SVM adalah 2, 3 dan 4 dengan metode klaster *k-means* dan *kernel k-means.* Pembagian data training testing adalah sebesar 80:20 dan proses klasifikasi diulang sebanyak 10 kali.
   3. Memperoleh metode klasifikasi terbaik dengan melihat rata-rata AUC.
   4. Melakukan analisis *survival* untuk masing-masing kategori nasabah (*early payment* dan *late payment*) dengan *cox proportional hazard* dan *survival* SVM.
   5. Memperoleh metode analisis *survival* terbaik dengan melihat nilai *C*-Indeks.
   6. Mendapatkan kesimpulan.

Berikut merupakan tahapan analisis untuk studi simulasi dan data riil.



Gambar 3.2 Tahapan Analisis Penelitian Data Simulasi



Gambar 3.3 Tahapan Analisis Penelitian Data Nasabah Gadai *Fintech*-X

BAB 4

ANALISIS DAN PEMBAHASAN

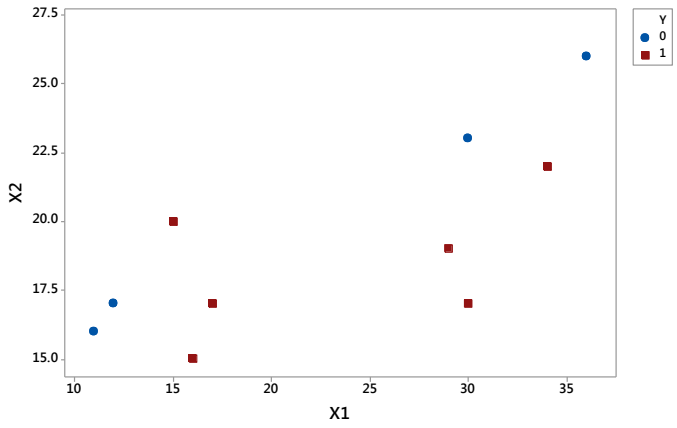
Pada bab ini berisi tentang uraian langkah-langkah untuk melakukan *ensemble* SVM. Kemudian dilanjutkan hasil performa pada studi simulasi berdasarkan skenario yang ada di Bab 3. Setelah itu dilakukan aplikasi pemodelan klasifikasi dan survival pada data nasabah fintech X.

4.1 Langkah-langkah *Clustered* SVM (*Ensemble* SVM)

Sebagai ilustrasi melakukan *ensemble* SVM diberikan data ilustrasi berupa variabel Y yang merupakan variabel respon dengan label (0,1). Kemudian untuk mempermudah ilustrasi secara visual, terdapat 2 variabel prediktor X_1 dan X_2. Variabel prediktor dibangkitkan secara sembarang dan variabel respon menyesuaikan bentuk visualisasi dengan batasan klaster yang terbentuk adalah 2. Berikut merupakan ilustrasi data yang akan dilakukan *ensemble* SVM.

Y	X_1	X_2
1	16	15
1	17	17
1	15	20
0	11	16
0	12	17
1	30	17
1	29	19
1	34	22
0	36	26
0	30	23

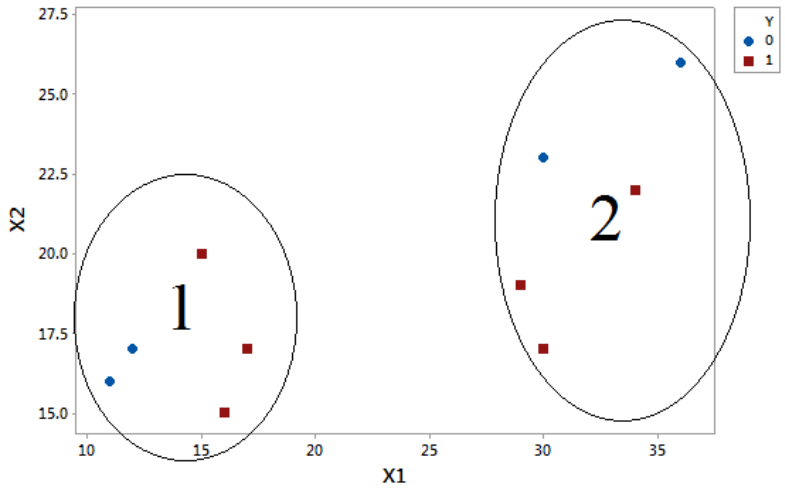
Visualisasi dari data tersebut dapat dilihat pada Gambar 4.1



Gambar 4.1 Visualisasi Data Ilustrasi Langkah *Ensemble* SVM

Langkah pertama memperoleh klaster untuk masing-masing data pengamatan. Pada kasus ini, banyak klaster yang digunakan adalah 2 dengan metode klaster adalah *k-means*. Sehingga diperoleh klaster untuk masing-masing pengamatan sebagai berikut

Y	X_1	X_2	Klaster
1	16	15	1
1	17	17	1
1	15	20	1
9	11	16	1
0	12	17	1
1	30	17	2
1	29	19	2
1	34	22	2
0	36	26	2
0	30	23	2



Gambar 4.2 Visualisasi Hasil Klaster Data Ilustrasi Langkah *Ensemble* SVM

Langkah kedua mendapatkan matrix transformasi $\tilde{\boldsymbol{x}}$

$$\tilde{\boldsymbol{x}} = \begin{bmatrix} 16 & 15 & 16 & 15 & 0 & 0 \\ 17 & 17 & 17 & 17 & 0 & 0 \\ 15 & 20 & 15 & 20 & 0 & 0 \\ 11 & 16 & 11 & 16 & 0 & 0 \\ 12 & 17 & 12 & 17 & 0 & 0 \\ 30 & 17 & 0 & 0 & 30 & 17 \\ 29 & 19 & 0 & 0 & 29 & 19 \\ 34 & 22 & 0 & 0 & 34 & 22 \\ 36 & 26 & 0 & 0 & 36 & 26 \\ 30 & 23 & 0 & 0 & 30 & 23 \end{bmatrix}$$

Langkah ketiga mendapatkan nilai $\tilde{\boldsymbol{w}}$ dengan batasan kondisi yang digunakan pada kasus ini adalah $\lambda = 1, C = 1$. Model SVM yang dibangun tanpa menggunakan *bias*. Sehingga nilai $\tilde{\boldsymbol{w}}$ pada persamaan (2.29) adalah sebagai berikut:

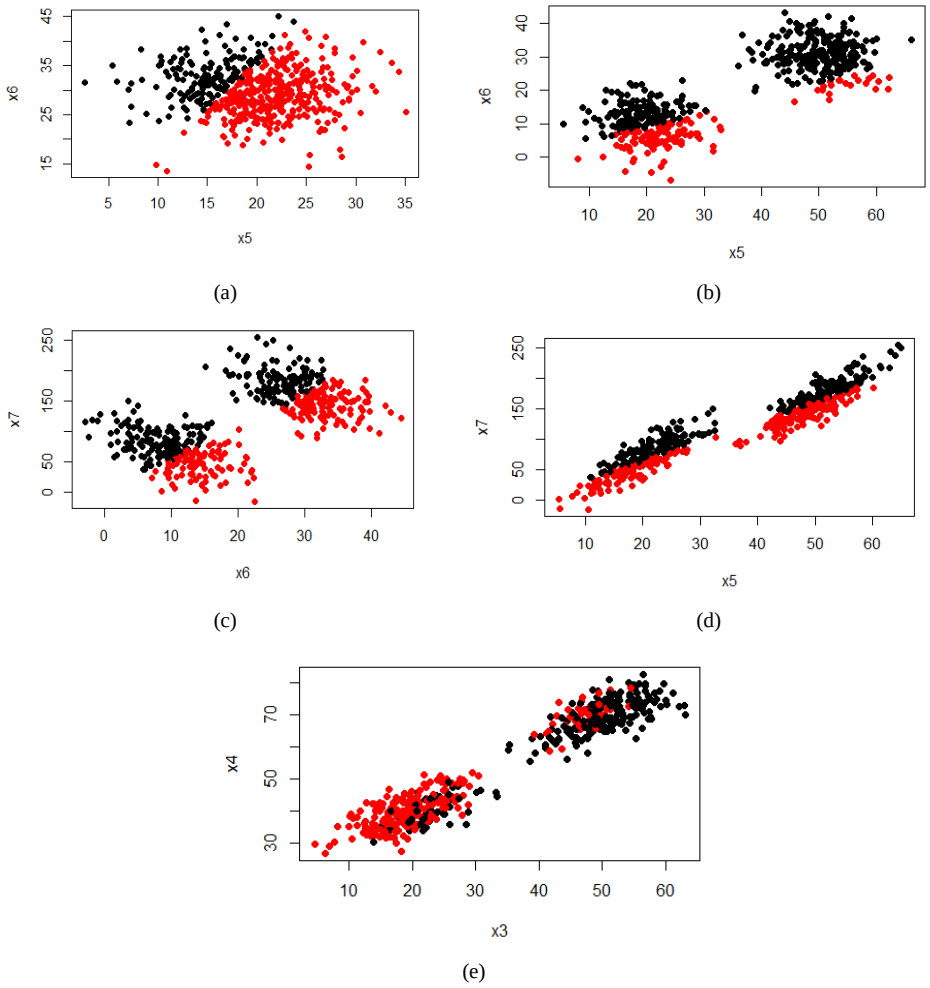
$$\tilde{\boldsymbol{w}}^T = [0.55 \quad -0.53 \quad 0.54 \quad -0.27 \quad 0.01 \quad -0.27]$$

Dua elemen pertama merupakan bobot untuk variabel X_1 dan X_2. Dua elemen berikutnya adalah bobot untuk variabel X_1 dan X_2 pada klaster 1. Dua elemen terakhir merupakan bobot untuk variabel X_1 dan X_2 pada klaster 2. Sehingga diperoleh $\tilde{\boldsymbol{x}}_i^T \tilde{\boldsymbol{w}}$ untuk setiap baris pada vektor $\tilde{\boldsymbol{x}}\tilde{\boldsymbol{w}}$ sebagai berikut:

$$\tilde{\boldsymbol{x}}\tilde{\boldsymbol{w}} = \begin{bmatrix} 5.29 \\ 4.76 \\ 0.16 \\ -0.95 \\ -0.67 \\ 2.99 \\ 0.81 \\ 1.17 \\ -0.94 \\ -1.86 \end{bmatrix}; sign(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{w}}) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}; \text{di mana } sign(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{w}}) = \begin{cases} 1, & \tilde{\boldsymbol{x}}\tilde{\boldsymbol{w}} \geq 0 \\ 0, & \tilde{\boldsymbol{x}}\tilde{\boldsymbol{w}} < 0 \end{cases}$$

4.2 Hasil Kajian Model Klasifikasi pada Data Simulasi

Dataset simulasi secara umum terbagi menjadi dua kategori yakni dataset yang mengandung variabel kategorik (skenario 1-5) dan dataset yang tidak mengandung variabel kategorik (skenario 6-10). Banyaknya variabel kategorik untuk masing-masing skenario adalah 4. Setelah membangkitkan data simulasi, dilakukan visualisasi pada variabel kontinu. Beberapa contoh visualisasi data kontinu dapat dilihat pada Gambar 4.3. Gambar 4.3 menunjukkan jika tidak terdapat efek *mixture* maka visualisasi dari *scatterplot* seperti pada gambar (a). Berdasarkan pengaturan skenario, terdapat 3 tipe *mixture* di variabel prediktor: (i) *mixture normal distribution*, (ii) kombinasi linier dari 2 variabel prediktor yang mengikuti *mixture normal distribution*, dan (iii) *mixture multivariate normal distribution*. Setiap efek *mixture* dapat dilihat pada Gambar 4.3 (b), (c), (d), and (e).



Gambar 4.3 Visualisasi Data Kontinu pada Beberapa Skenario Simulasi, (a) Skenario 1, (b) Skenario 2, (c) Skenario 3, (d) Skenario 4, and (e) Skenario 8.

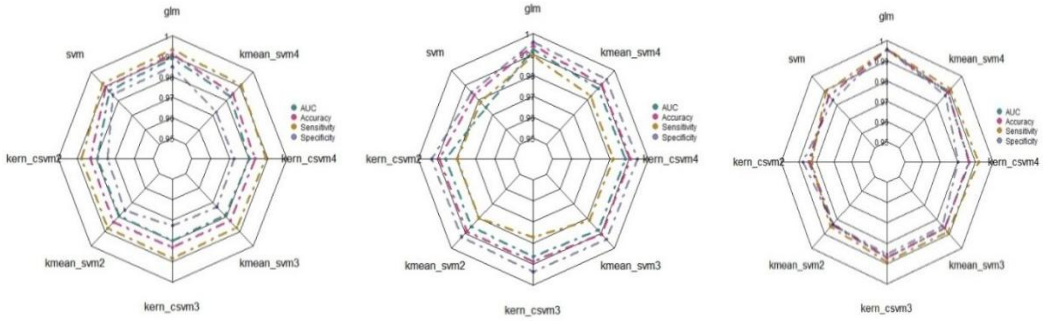
Label pada variabel respon dibangkitkan berbasis model regresi logistik. Metode klasifikasi yang digunakan adalah regresi logistik, SVM dan *Ensemble* SVM (*Ensemble* SVM). Mengingat diketahui bahwa label yang sebenarnya dihasilkan dari model logit, kita ingin tahu bagaimana SVM dan *Ensemble* SVM dapat mengkompensasi regresi logistik dengan kondisi prediktor mengikuti pengaturan simulasi. Kriteria kebaikan model yang digunakan adalah *Area Under Curve* ROC (AUC), akurasi, sensitivitas, spesifisitas. Adapun parameter SVM σ yang digunakan sebesar 0,7 dan pada *Ensemble* SVM $\lambda = 1$.

Gambar 4.4 dan Gambar 4.5 menunjukkan bahwa regresi logistik memiliki peforma terbaik. Hal ini wajar karena nilai variabel respon yang dibangkitkan mengikuti model regresi logistik. Namun, SVM dan *Ensemble* SVM juga mampu

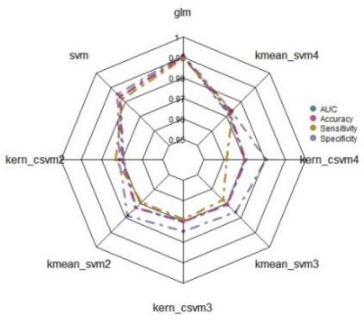
mengimbangi peforma regresi logistik dalam beberapa skenario. Metode *k-mean* dan *kernel k-means* pada *Ensemble* SVM menghasilkan peforma yang sama. Peningkatan jumlah klaster tidak memberikan efek yang signifikan karena data *mixture* yang dibangun hanya dua.

Hasil pada studi simulasi untuk skenario 1 menunjukkan bahwa SVM bekerja sebaik regresi logistik. Sedangkan penggunaan *Ensemble* SVM menurunkan akurasi dan ukuran lainnya. Dalam skenario dua, ketika pola *mixture* ada dalam prediktor, *Ensemble* SVM meningkatkan kinerja SVM. *Ensemble* SVM melakukan hampir sama baiknya dengan regresi logistik, kecuali untuk *sensitivity*. Pada kondisi multikolinieritas dan kombinasi linier antara variabel prediktor terdapat penurunan peformansi di model *Ensemble* SVM. Sementara itu, SVM masih memiliki performa yang baik. Selisih peformansi dengan regresi logistik cukup kecil. SVM bahkan berkinerja sebaik regresi logistik dalam skenario lima.

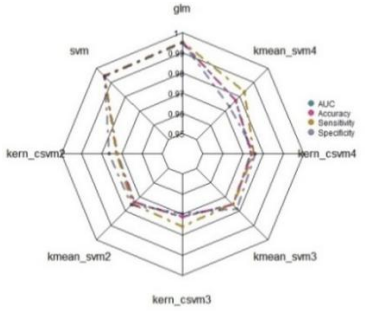
Anehnya dalam kondisi ketika semua prediktor bersifat kontinu (skenario 6-10), peforma *Ensemble* SVM berkurang banyak, sedangkan SVM masih menunjukkan hasil yang cukup baik. Namun jika kita membandingkan interval kepercayaan kriteria evaluasi antara regresi logistik dan SVM, maka dapat disimpulkan bahwa kinerja keduanya tidak berbeda. Hasil interval kepercayaan untuk masing-masing skenario dapat dilihat *box plot* yang tersedia pada Lampiran 12 s.d. 21



(a) (b) (c)

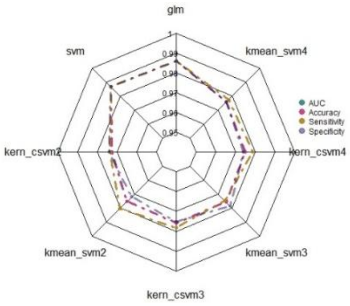


(d)

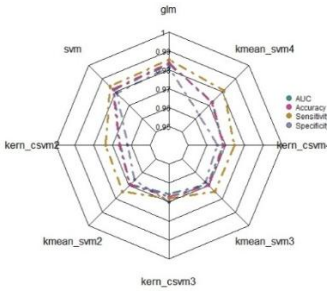


(e)

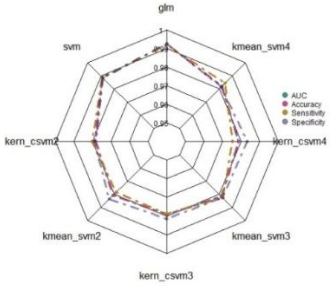
Gambar 4.4 Peforma Model Klasifikasi (*Average Value*), (a) Skenario 1, (b) Skenario 2, (c) Skenario 3, (d) Skenario 4, and (e) Skenario 5.



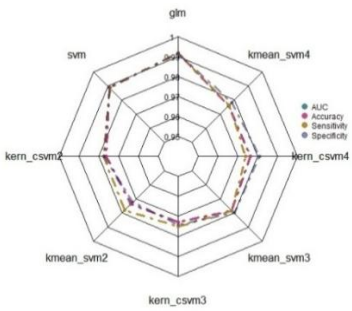
(a)



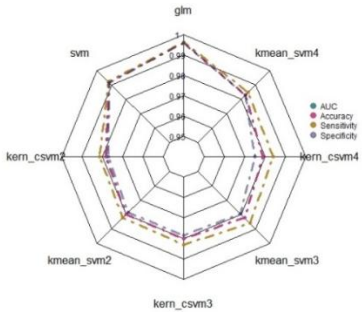
(b)



(c)



(d)



(e)

Gambar 4.5 Peforma Model Klasifikasi (*Average Value*), (a) Skenario 6, (b) Skenario 7, (c) Skenario 8, (d) Skenario 9, and (e) Skenario 10.

Walaupun peforma regresi logistik unggul disemua kondisi namun terdapat nilai estimasi parameter yang berbeda dari parameter yang sesungguhnya. Tabel 4.1 merupakan hasil estimasi parameter beserta uji signifikansi untuk skenario 1 sampai 5 pada salah satu iterasi. Dapat dilihat bahwa ketika terdapat skenario kombinasi linier antar variabel (skenario 4 dan 5), hasil estimasi parameter tidak dapat ditampilkan karena proses yangtidak konvergen. Kemudian hasil estimasi parameter untuk variabel prediktor penyusun kombinasi linier semakin menjauhi

nilai sesungguhnya dan berubah tanda. Sebagai contoh pada skenario 5, variabel prediktor penyusun kombinasi linier terdiri dari X_5 dan X_6. Nilai koefisien yang sebenarnya masing-masing sebesar 13 dan -3. Namun ketika dilakukan estimasi diperoleh hasil -15 dan 25. Selain itu hasil estimasi parameter yang dihasilkan juga signifikan. Hal ini cukup berbahaya apabila terjadi pada kasus data riil karena dapat menyebabkan kesalahan interpretasi model. Akan tetapi lain halnya bila variabel prediktor mengikuti distribusi normal multivariat. Dapat dilihat bahwa tidak terjadi kondisi perubahan tanda dalam proses estimasi. Pada salah satu skenario yakni skenario 3 terlihat bahwa walaupun variabel X_7 dan X_8 saling berkorelasi namun hasil estimasi parameter yang dihasilkan mendekati nilai sesungguhnya.

Pada skenario 6 sampai 10 juga terjadi hal yang sama. Tabel 4.2 menunjukkan ketika terdapat variabel prediktor yang merupakan kombinasi linier (X_3) antar prediktor maka estimasi parameter pada variabel tersebut tidak dapat muncul dan variabel penyusun kombinasi linier (X_1 dan X_2) memiliki estimasi parameter yang berbeda jauh dari nilai sesungguhnya. Selain itu terjadi perubahan tanda dan variabel tersebut juga signifikan. Selanjutnya pada variabel yang dibangkitkan dengan distirbusi normal multivariat juga menghasilkan kondisi yang sama seperti pada skenario sebelumnya. Dan juga dari skenario 2-5 dan 7-10 ternyata efek mixture tidak berpengaruh terhadap proses estimasi parameter regresi. Hasil kecuali pada skenario yang mengandung kombinasi linier estimasi untuk prediktor yang berdistribusi *mixture* mendekati nilai sesungguhnya.

Tabel 4.1. Estimasi Parameter Regresi Logistik pada Skenario 1 sampai 5

Skenario	Variabel	Koefisien	$\widehat{\text{Koefisien}}$	Std.Error	Z	P Value
1	X_1	2	3.777	1.861	2.030	0.042
	X_2	4	2.788	1.737	1.604	0.109
	X_3	3	0.479	2.354	0.204	0.839
	X_4	0	-3.883	2.262	-1.716	0.086
	X_5	7	8.799	2.750	3.200	0.001
	X_6	-4	-4.981	1.559	-3.194	0.001
2	X_1	2	1.608	1.767	0.910	0.363
	X_2	4	1.636	1.446	1.131	0.258
	X_3	3	5.923	2.201	2.691	0.007
	X_4	0	1.236	1.436	0.861	0.389
	X_5	4	4.336	1.255	3.456	0.001
	X_6	-9	-9.712	2.807	-3.460	0.001

Tabel 4.1 Estimasi Parameter Regresi Logistik pada Skenario 1 sampai 5 (Lanjutan)

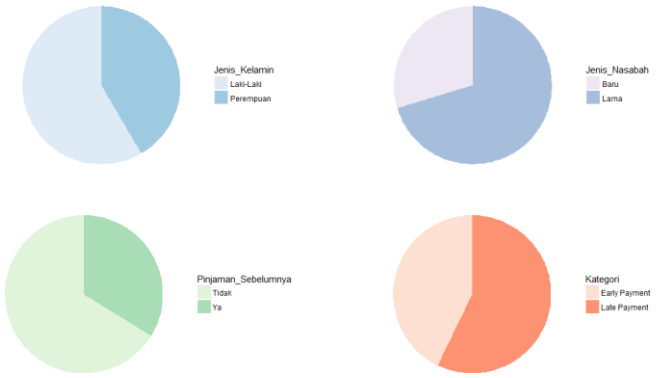
Skenario	Variabel	Koefisien	$\widehat{\text{Koefisien}}$	Std.Error	Z	P Value
3	X_1	2	1.024	2.061	0.497	0.619
	X_2	4	2.408	2.117	1.137	0.255
	X_3	3	2.802	2.403	1.166	0.244
	X_4	0	0.954	1.586	0.601	0.548
	X_5	4	3.101	1.231	2.518	0.012
	X_6	-4	-3.149	1.244	-2.531	0.011
	X_7	8	6.548	2.559	2.559	0.011
	X_8	-6	-4.819	1.890	-2.549	0.011
4	X_1	2	6.730	3.878	1.736	0.083
	X_2	4	5.185	3.292	1.575	0.115
	X_3	3	14.108	8.883	1.588	0.112
	X_4	0	-5.122	3.542	-1.446	0.148
	X_5	13	-15.201	8.913	-1.705	0.088
	X_6	-3	25.451	14.929	1.705	0.088
	X_7	3.5	NA	NA	NA	NA
5	X_1	2	7.841	4.953	1.583	0.113
	X_2	4	7.868	4.617	1.704	0.088
	X_3	3	3.363	2.289	1.469	0.142
	X_4	0	4.030	3.511	1.148	0.251
	X_5	13	-7.780	3.109	-2.503	0.012
	X_6	-3	12.865	5.143	2.501	0.012
	X_7	-3.5	NA	NA	NA	NA
	X_8	5.5	9.350	3.839	2.436	0.015
	X_9	-3	-5.084	2.089	-2.433	0.015

Tabel 4.2. Estimasi Parameter Regresi Logistik pada Skenario 6 sampai 10

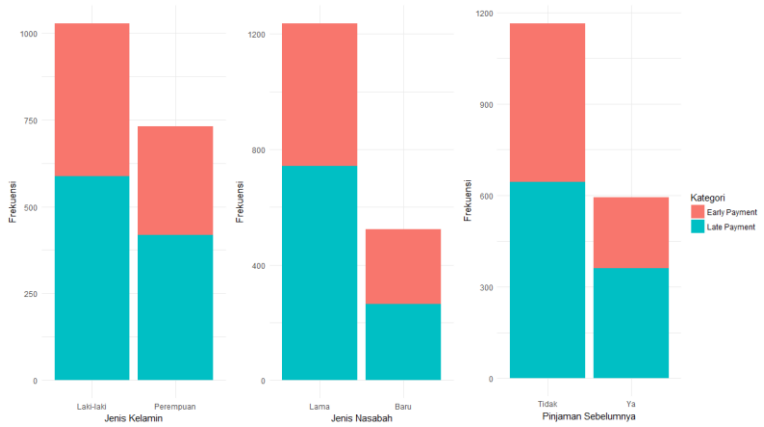
Skenario	Variabel	Koefisien	$\widehat{\text{Koefisien}}$	Std.Error	Z	P Value
6	X_1	6	7.795	2.366	3.294	0.001
	X_2	-4	-5.195	1.574	-3.300	0.001
7	X_1	3	4.173	1.021	4.088	0.000
	X_2	-5	-6.964	1.704	-4.086	0.000
8	X_1	4	4.727	1.716	2.755	0.006
	X_2	-5	-5.926	2.119	-2.797	0.005
	X_3	-6	-7.117	2.602	-2.735	0.006
	X_4	3	3.591	1.305	2.753	0.006
9	X_1	4	-5.184	1.596	-3.249	0.001
	X_2	4.5	8.983	2.771	3.242	0.001
	X_3	-2	NA	NA	NA	NA
10	X_1	6	-14.043	6.924	-2.028	0.043
	X_2	6	23.456	11.539	2.033	0.042
	X_3	-3	NA	NA	NA	NA
	X_4	5	7.675	3.856	1.990	0.047
	X_5	-2	-3.060	1.532	-1.997	0.046

4.2 Hasil Kajian Model Klasifikasi pada Data Nasabah Gadai *Fintech*-X

Sebelum dilakukan pemodelan klasifikasi, dilakukan analisis eksplorasi data untuk masing-masing variabel. Gambar 4.6 menunjukkan bahwa proporsi jenis kelamin nasabah, kategori nasabah dan apakah nasabah memiliki pinjaman sebelumnya memiliki proporsi yang cenderung sama. Kemudian mayoritas nasabah yang melakukan gadai merupakan nasabah lama. Apabila ditinjau dari kategori nasabah, berdasarkan Gambar 4.7 proporsi nasabah pada masing-masing variabel jenis kelamin, jenis nasabah, dan kepemilikan pinjaman sebelumnya cenderung sama untuk masing-masing kategori baik itu *early payment* maupun *late payment*.



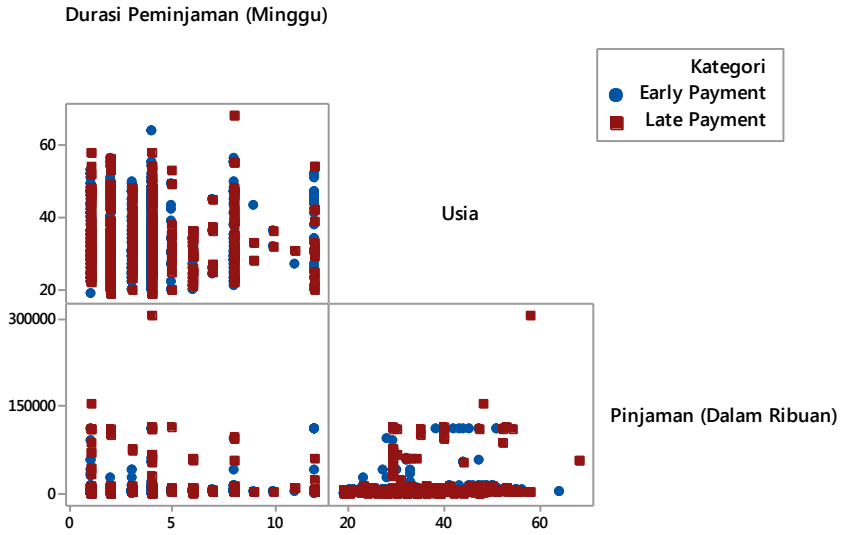
Gambar 4.6 Karakteristik Data Nasabah Gadai *Fintech*-X pada Variabel X_4, X_5, X_6, dan Y



Gambar 4.7 Karakteristik Data Nasabah Gadai *Fintech*-X pada Variabel X_4, X_5, X_6, berdasarkan Kategori Nasabah

Pada variabel kontinu (Usia, Besaran Pinjaman, dan Durasi Peminjaman), *matrix scatterplot* pada Gambar 4.8 menunjukkan bahwa tidak ada pola khusus

yang dapat memisahkan antara nasabah *early payment* maupun *late payment*. Terlihat bahwa titik-titik untuk setiap kategori saling menumpuk.



Gambar 4.8 Karakteristik Data Nasabah Gadai *Fintech*-X pada Variabel X_1, X_2, dan X_3

Pemodelan klasifikasi pada data riil menggunakan regresi logistik, SVM dan *Ensemble* SVM. Proses *training* dan *testing* data menggunakan proporsi sebesar 80:20. *Cross Validation* yang digunakan adalah *reapeated hold-out* dengan replikasi sebanyak 10 kali. Optimasi parameter pada SVM dan Ensemble SVM menggunakan metode *grid search.* Grid yang digunakan untuk parameter SVM yakni $C = \{10{,}20, \dots, 100\}$ dan $\sigma = \{1{,}2, \dots, 10\}$. Sedangkan *Ensemble SVM* $C = \{10{,}20, \dots, 100\}$ dan $\lambda = \{1{,}2, \dots, 10\}$. Kriteria kebaikan model yang digunakan pada penelitian kali ini adalah AUC.

Terdapat 7 tipe model klasifikasi yang dibangun. Rincian tipe model klasifikasi sebagai berikut:

- Model 1: Semua observasi dengan menggunakan keseluruhan Variabel Prediktor.
- Model 2: Semua observasi dengan menggunakan Variabel Prediktor yang kontinu.
- Model 3: Pinjaman terakhir nasabah dengan menggunakan keseluruhan Variabel Prediktor.

- Model 4: Pinjaman terakhir nasabah dengan menggunakan Variabel Prediktor yang kontinu.
- Model 5: Menggunakan data nasabah baru, Sehingga berdampak tereliminasinya variabel X_5 dan X_6
- Model 6: Semua observasi dengan menggunakan keseluruhan Variabel Prediktor. Namun kode variabel kategorik X_6 diubah menjadi, 0=memiliki pinjaman sebelumnya dan 1=tidak memiliki.
- Model 7: Pinjaman terakhir nasabah dengan menggunakan keseluruhan Variabel Prediktor. Namun kode variabel kategorik X_6 diubah menjadi, 0=memiliki pinjaman sebelumnya dan 1=tidak memiliki.

Pemilihan skenario model yang akan dibangun seperti di atas didasari oleh batasan asumsi pada regresi logistik. Beberapa asumsi regresi logistik adalah independen antar observasi pengamatan dan tidak adanya multikolinieritas antar variabel prediktor. Pada data nasabah gadai *Fintech* X terdapat indikasi kedua asumsi tersebut tidak dipenuhi. Proses gadai suatu nasabah dapat dilakukan lebih dari satu kali walaupun nasabah masih belum melunasi tanggungan gadai yang sebelumnya. Sehingga terdapat dependensi antar observasi. Selain itu pada variabel X_5 dan X_6 terdapat indikasi multikolinieritas hal tersebut terjadi karena ketika nasabah merupakan nasabah baru tentunya pada variabel kepemilikan tanggungan sebelumnya (X_6) nasabah berstatus tidak memiliki tanggungan. Oleh sebab itu skenario model 1, 2, dan 6 merupakan kondisi dimodelkan secara paksa apabila kedua asumsi tersebut dilanggar khususnya untuk independen antar observasi. Skenario 3, 4, dan 7 merupakan kondisi apabila dependensi antar observasi dihilangkan. Skenario 5 merupakan kondisi apabila dependensi dan multikolinieritas dihilangkan.

Pemodelan non parametrik dilakukan terlebih dahulu untuk mengetahui informasi terkait variabel yang signifikan terhadap model. Metode parametrik yang digunakan pada penelitian ini adalah regresi logistik. Model yang dibangun tanpa melibatkan intersep β_0. Hasil estimasi dan signifikansi untuk masing-masing variabel dan skenario model pada salah satu iterasi dapat dilihat pada Tabel 4.3.

Tabel 4.3. Estimasi Parameter dan Pengujian Signifikansi Parameter Regresi Logistik pada Data Nasabah Gadai *Fintech* - X

Model	Variabel	Koefisien	Exp(Koefisien)	Std.Error	Z	*P Value*
1	X_1	0.008	1.009	0.004	2.357	0.018
	X_2	0.000	1.000	0.000	-0.357	0.721
	X_3	0.012	1.012	0.022	0.535	0.593
	X_4	0.035	1.036	0.108	0.328	0.743
	X_5	-0.395	0.673	0.128	-3.091	0.002
	X_6	0.112	1.119	0.125	0.897	0.370
2	X_1	0.007	1.007	0.003	2.455	0.014
	X_2	0.000	1.000	0.000	-0.433	0.665
	X_3	0.012	1.012	0.022	0.537	0.591
3	X_1	0.004	1.005	0.007	0.685	0.493
	X_2	0.000	1.000	0.000	1.841	0.066
	X_3	-0.072	0.930	0.041	-1.785	0.074
	X_4	-0.163	0.850	0.208	-0.782	0.434
	X_5	-1.430	0.239	0.265	-5.401	0.000
	X_6	-0.218	0.804	0.230	-0.951	0.342
4	X_1	-0.011	0.989	0.005	-2.330	0.020
	X_2	0.000	1.000	0.000	1.993	0.046
	X_3	-0.072	0.931	0.039	-1.858	0.063
5	X_1	0.053	1.054	0.093	0.567	0.570
	X_2	-0.112	0.894	0.104	-1.079	0.281
	X_3	0.001	1.001	0.029	0.034	0.973
	X_4	0.106	1.112	0.163	0.652	0.514
6	X_1	0.008	1.008	0.004	2.148	0.032
	X_2	0.000	1.000	0.000	0.437	0.662
	X_3	0.007	1.007	0.022	0.298	0.766
	X_4	0.047	1.048	0.108	0.431	0.666
	X_5	-0.303	0.739	0.133	-2.274	0.023
	X_6	0.027	1.028	0.125	0.218	0.828
7	X_1	0.008	1.008	0.007	1.228	0.219
	X_2	0.000	1.000	0.000	2.029	0.042
	X_3	-0.086	0.918	0.039	-2.200	0.028
	X_4	-0.128	0.880	0.206	-0.622	0.534
	X_5	-1.210	0.298	0.275	-4.402	0.000
	X_6	-0.268	0.765	0.229	-1.171	0.242

Berdasarkan Tabel 4.3 diperoleh hasil bahwa variabel yang cukup konsisten signifikan pada masing-masing skenario model adalah usia nasabah (X_1) dan jenis nasabah (X_5). Sedangkan variabel yang cukup konsisten tidak signifikan adalah

jenis kelamin (X_4). Kemudian pada skenario model 6 dan 7 terjadi keanehan yakni tidak ada perubahan tanda pada parameter. Seharusnya jika dibandingkan pada skenario model 1 dan 3 tanda pada variabel X_6 berubah. Namun hal tersebut masih dapat ditoleransi karena variabel tersebut tidak signifikan. Model bentuk transformasi logit $g(\boldsymbol{X})$ untuk masing-masing skenario disajikan sebagai berikut

$$g_1(\boldsymbol{X}) = 0.008X_1 + 0.000X_2 + 0.012X_3 + 0.035X_4 - 0.395X_5 + 0.112X_6$$

$$g_2(\boldsymbol{X}) = 0.007X_1 + 0.000X_2 + 0.012X_3$$

$$g_3(\boldsymbol{X}) = 0.004X_1 + 0.000X_2 - 0.072X_3 - 0.163X_4 - 1.430X_5 - 0.218X_6$$

$$g_4(\boldsymbol{X}) = -0.011X_1 + 0.000X_2 - 0.072X_3$$

$$g_5(\boldsymbol{X}) = 0.053X_1 - 0.112X_2 + 0.001X_3 + 0.106X_4$$

$$g_6(\boldsymbol{X}) = 0.008X_1 + 0.000X_2 + 0.007X_3 + 0.047X_4 - 0.303X_5 + 0.027X_6$$

$$g_7(\boldsymbol{X}) = 0.008X_1 + 0.000X_2 - 0.086X_3 - 0.128X_4 - 1.210X_5 - 0.268X_6$$

Setiap persamaan bentuk logit dapat dimasukkan kedalam persamaan regresi logistik seperti pada persamaan 2.3. Interpretasi model regresi logistik pada variabel signifikan sebagai ilustrasi untuk skenario model 1 pada variabel X_5 adalah apabila nasabah baru memiliki resiko $\exp(-0.395)$ atau 0.673 kali mengalami keterlambatan pembayaran (*late payment*) dibandingkan dengan nasabah lama.

Permodelan yang selanjutnya dilakukan dengan metode non paramtrik yakni dengan SVM dan *Ensemble* SVM. Hasil evaluasi model klasifikasi pada data nasabah Fintech X dapat dilihat pada Tabel 4.4 dan 4,5.

Tabel 4.4. Rata-Rata AUC Model Klasifikasi pada Data Nasabah *Fintech*-X untuk Model 1 sampai 4

Metode	Model 1		Model 2		Model 3		Model 4	
	Train	Test	Train	Test	Train	Test	Train	Test
SVM	0.8728	0.5440	0.7146	0.5594	**0.9472**	0.5547	0.7933	0.7933
Kmean_CSVM2	0.5318	0.5401	0.5133	0.5165	0.5058	0.5057	0.5058	0.5058
Kern_CSVM2	0.5333	0.5366	0.5120	0.5121	0.5063	0.5069	0.5063	0.5063
Kmean_CSVM3	0.5435	0.5363	0.5214	0.5168	0.5081	0.5065	0.5081	0.5081
Kern_CSVM3	0.5421	0.5592	0.5354	0.5288	0.4931	0.5022	0.4931	0.4931
Kmean_CSVM4	0.5445	0.5350	0.5284	0.5219	0.5081	0.5100	0.5081	0.5081
Kern_CSVM4	0.5154	0.5110	0.4827	0.4863	0.5000	0.5000	0.5000	0.5000
Regresi Logistik	0.5268	0.5311	0.5005	0.4961	0.5165	0.5092	0.5120	0.5120

Tabel 4.5. Rata-Rata AUC Model Klasifikasi pada Data Nasabah *Fintech*-X untuk Model 5 sampai 7

Metode	Model 5		Model 6		Model 7	
	Train	**Test**	**Train**	**Test**	**Train**	**Test**
SVM	0.8977	0.5517	0.8957	0.5481	0.9453	**0.5596**
Kmean_CSVM2	0.5553	0.5392	0.5007	0.5005	0.5058	0.5057
Kern_CSVM2	0.5483	0.5551	0.5005	0.5001	0.5063	0.5069
Kmean_CSVM3	0.5645	0.5330	0.5008	0.5025	0.5081	0.5065
Kern_CSVM3	0.5319	0.5426	0.5003	0.5017	0.4931	0.5022
Kmean_CSVM4	0.5691	0.5454	0.5012	0.5067	0.5081	0.5099
Kern_CSVM4	0.4912	0.5021	0.5000	0.5000	0.5000	0.5000
Regresi Logistik	0.5165	0.5092	0.5240	0.5242	0.5314	0.5213

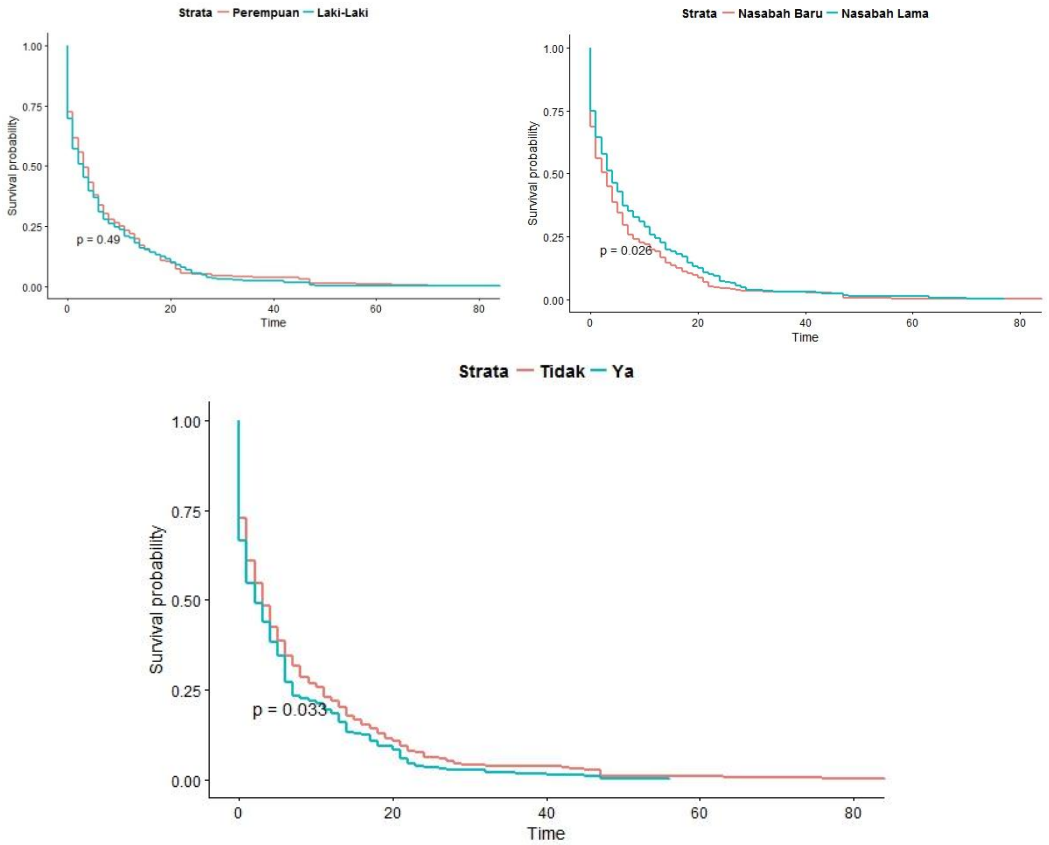
Berdasarkan Tabel 4.4 dan 4.5, SVM memberikan peforma paling baik. Namun pada setiap skenario model terjadi *overfitting*. Penambahan jumlah klaster dan metode klaster pada *Ensemble* SVM tidak memberikan pengaruh signifikan terhadap peningkatan peforma. Walaupun demikian metode terbaik yang dihasilkan masih *out of perform*. Hal tersebut disebabkan rata-rata AUC pada model dengan parameter terbaik masih berada di sekitar 0.5. Sehingga dapat dikatakan variabel dan metode klasifikasi yang digunakan belum dapat mendiskriminasi dengan baik.

4.3 Pemodelan Analisis *Survival* dengan *Cox Proportional Hazard* dan *Survival* SVM pada Nasabah Gadai *Fintech*-X

Pemodelan analisis survival terbagi menjadi dua yakni untuk nasabah *early payment* dan *late payment*. Setiap kriteria nasabah dimodelkan menggunakan *Cox Proportional Hazard* dan *Survival* SVM. Kriteria kebaikan metode yang digunakan adalah *C-Index*.

4.3.1 Pemodelan Analisis Survival Nasabah *Early Payment*

Kurva *Kaplan Meier* dan Uji *Log Rank* dilakukan terlebih dahulu sebelum memodelkan dengan *Cox Proportional Hazard* dan *Survival* SVM. Kurva *Kaplan Meier* dan Uji *Log Rank* dilakukan untuk mengetahui perbedaan kurva *survival* antara variabel kategorik yang digunakan pada penelitian ini. Hasil Kurva *Kaplan Meier* dan Uji *Log Rank* disajikan pada Gambar 4.9.



Gambar 4.9 Kurva *Kaplan Meier* dan *P-value* Uji *Log Rank* pada Nasabah *Early Payment*

Berdasarkan Gambar 4.9 diperoleh bahwa kurva *survival* pada nasabah laki-laki dan perempuan tidak ada perbedaan. Sedangkan pada nasabah baru dengan nasabah lama dan nasabah yang memiliki tanggungan sebelumnya terdapat perbedaan kurva *survival*. Terlihat pada nasabah lama cenderung memiliki peluang lebih cepat dalam melunasi barang yang telah digadaikan. Sedangkan pada nasabah yang tidak memiliki tanggungan pelunasan barang yang digadai juga cenderung memiliki peluang melunasi lebih cepat. Selain secara visual kondisi tersebut didukung dari nilai *p-value* uji *log-rank*. Pada variabel jenis nasabah dan kepemilikan tanggungan, *p-value* masing-masing uji log-rank sebesar 0.026 dan 0.033. Karena *p-value* kurang dari $\alpha = 0.05$ maka keputusannya adalah tolak H_0. Yakni ada perbedaan kurva yang signifikan antara jenis nasabah baru dan lama serta terdapat perbedaan yang signifikan pula antara nasabah yang memiliki tanggungan sebelumnya dengan yang tidak memiliki tanggungan sebelumnya

Analisis dilanjutkan dengan melakukan pemodelan *Cox Proportional Hazard*. Pada pemodelan *Cox Proportional Hazard* terdapat dua hal yang dilakukan yakni pengujian parsial untuk masing-masing variabel prediktor dan pengujian asumsi *Cox Proportional Hazard*. Hasil estimasi parameter dan pengujian parsial disajikan pada Tabel 4.6.

Tabel 4.6 Cox PH pada Nasabah *Early Payment*

Variabel	Koefisien	Exp(Koefisien)	S.E. (Koefisien)	Wald	P-value
X_1	0.043	1.044	0.037	1.150	0.250
X_2	-0.059	0.942	0.044	-1.356	0.175
X_3	-0.116	0.890	0.019	-6.213	0.000
X_4	0.056	1.058	0.075	0.747	0.455
X_5	-0.181	0.835	0.078	-2.317	0.021
X_6	0.079	1.082	0.075	1.055	0.291

Berdasarkan Tabel 4.6, variabel durasi peminjaman (X_3) dan jenis nasabah (X_5) berpengaruh signifikan terhadap laju pembayaran untuk nasabah *early payment* di perusahaan *Fintech-X*. Sedangkan variabel lainnya tidak berpengaruh signifikan. Nilai *odds ratio* pada variabel yang signifikan dapat dilihat pada hasil $\exp(koef)$. Misalnya pada variabel jenis nasabah, diperoleh nilai *odds ratio* sebesar 0.835 yang artinya nasabah baru memiliki kemungkinan 0.835 kali lebih cepat dalam melakukan pembayaran dibandingkan dengan nasabah lama. Selanjutnya model regresi *Cox Proportional Hazard* dapat dituliskan sebagai berikut.

$$h(t,\boldsymbol{x})_{ep} = h_0(t)\exp(0.043X_1 - 0.059X_2 - 0.116X_3 + 0.056X_4 - 0.181X_5 + 0.079X_6)$$

Tabel 4.7 Pengujian Asumsi Cox PH pada Nasabah *Early Payment*

Variabel	Korelasi	P-Value
X_1	-0.003	0.935
X_2	0.008	0.819
X_3	-0.108	0.000
X_4	-0.047	0.191
X_5	0.033	0.363
X_6	-0.023	0.516

Akan tetapi pemodelan Cox Proportional Hazard memiliki kelemahan yakni asumsi proportional hazard yang harus dipenuhi. Pada Tabel 4.7 dapat dilihat bahwa variabel durasi peminjaman tidak memenuhi asumsi. Karena model *Cox Proportional Hazard* tidak memenuhi asumsi, maka dilakukan pemodelan lain yang tidak memperhatikan asumsi. Pada penelitian ini metode yang digunakan untuk mengatasi permasalahan tersebut adalah dengan *Survival* SVM.

Terdapat 2 parameter γ dan μ serta parameter kernel RBF σ pada model Survival SVM. Pada penelitian ini nilai parameter yang digunakan masing-masing adalah 0.1, 0.3 dan 1. Hasil statistika deskriptif dari prognostik indeks yang dihasilkan adalah sebagai berikut.

Tabel 4.8 Statistika Deskriptif Prognostik Indeks *Survival* SVM Nasabah *Early Payment*

Variabel	Rata-Rata	Median	Standar Deviasi
Prognostik Indeks	7.1740	7.1233	0.3746

Prognostik indeks dari nasabah dapat dikategorikan ke dalam dua kategori yaitu nasabah yang memliki peluang yang besar untuk semakin lama melunasi hutangnya (*high risk*) dan nasabah yang peluangnya kecil untuk semakin lama melunasi hutangnya (*low risk*). Pengategorian didasarkan oleh nilai rata-rata prognostik indeks. Apabila prognostik indeks seorang nasabah di atas rata-rata maka nasabah tersebut masuk dalam kategori *high risk* dan sebaliknya. Namun pada kasus *early payment* tidak ada kategori seperti itu. Dikarenakan nasabah *early payment* sangat menguntungkan perusahaan. Proses kategori tersebut lebih bermakna untuk nasabah *late payment.*

Setelah diperoleh nilai prognostik untuk setiap observasi, maka dapat dihitung performa model Survival SVM maupun *Cox Proportional Hazard*. Berdasarkan Tabel 4.9, metode *Survival* SVM jauh lebih baik dari *Cox Proportional Hazard*.

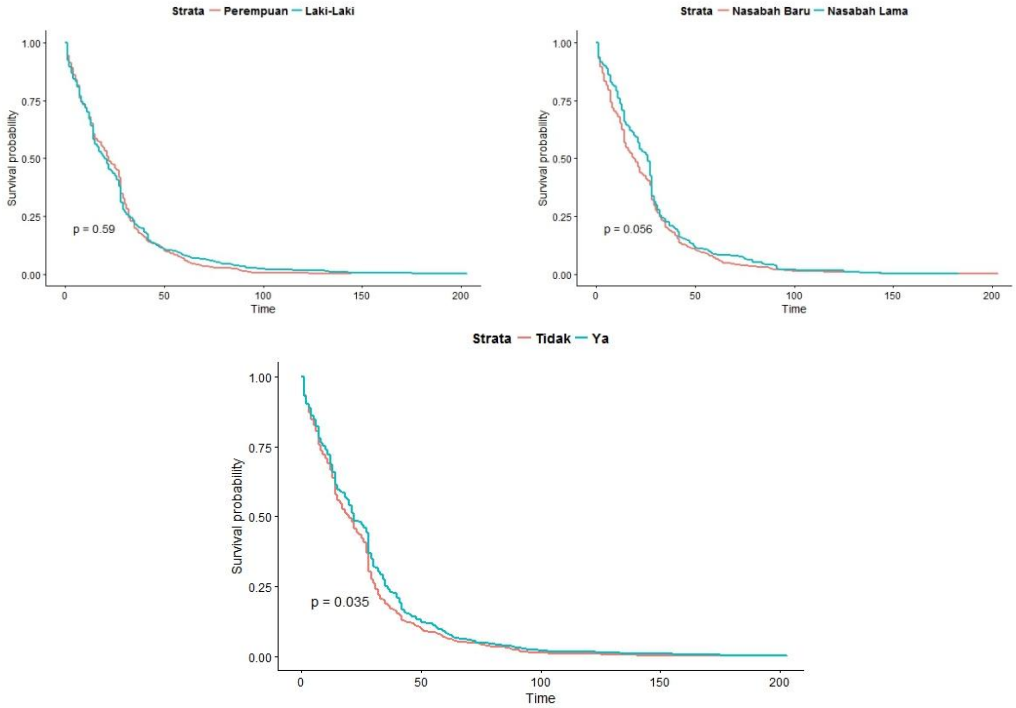
Tabel 4.9 C-Index Masing-Masing Metode untuk Nasabah *Early Payment*

Metode	C-Index
Cox PH	0.4186
Survival SVM	0.7193

4.3.2 Pemodelan Analisis Survival Nasabah *Late Payment*

Pemodelan analisis survival nasabah *late payment* sebelumnya juga dilakukan terlebih dahulu analisis Kurva Kaplan Meier dan uji log rank. Hasil

Kurva Kaplan Meier dan Uji Log Rank untuk nasabah *late payment* disajikan pada Gambar 4.10.



Gambar 4.10 Kurva *Kaplan Meier* dan *P-Value* Uji *Log Rank* pada Nasabah *Late Payment*

Berdasarkan Gambar 4.10 diperoleh bahwa kurva survival pada nasabah laki-laki dan perempuan tidak ada perbedaan. Sedangkan pada nasabah baru dengan nasabah lama dan nasabah yang memiliki tanggungan sebelumnya terdapat perbedaan kurva survival. Terlihat pada T 0-30 hari nasabah lama cenderung memiliki peluang lebih lama dalam melunasi barang yang telah digadaikan. Namun setelah 30 hari nasabah lama maupun baru cenderung memiliki peluang yang sama dalam melunasi barang yang telah digadaikan. Selanjutnya, pada nasabah yang memiliki tanggungan pelunasan barang yang digadai, saat T 20-70 hari cenderung memiliki peluang melunasi lebih lama dibandingkan nasabah yang tidak memiliki tanggungan sebelumnya. Hal tersebut diduga karena terdapat beberapa nasabah yang cenderung ketika melunasi tanggungan diikutkan pada pelunasan tanggungan yang terakhir. Sebagai contoh pada Tabel 4.10 berisi tentang tanggal peminjaman, jatuh tempo dan pelunasan dari beberapa nasabah dalam melunasi tanggungannya. Walaupun begitu, secara pengujian menggunakan uji log-rank variabel kepemilikan

tanggungan sebelumnya memiliki perbedaan kurva yang signifikan. Hal tersebut dapat dilihat dari *p-value* uji log rank yang lebih kecil dari $\alpha = 0.05$

Tabel 4.10 Periode Peminjaman, Jatuh Tempo dan Pelunasan Beberapa Nasabah

ID	Tanggal Peminjaman	Tanggal Jatuh Tempo	Tanggal Pelunasan
184	5/17/2016	6/14/2016	7/14/2016
184	6/13/2016	7/12/2016	7/14/2016
248	6/16/2015	8/11/2015	9/8/2015
248	8/10/2015	9/8/2015	9/8/2015
483	4/22/2016	5/6/2016	6/3/2016
483	5/4/2016	6/3/2016	6/3/2016

Analisis dilanjutkan dengan melakukan pemodelan *Cox Proportional Hazard*. Pada pemodelan *Cox Proportional Hazard* terdapat dua hal yang dilakukan yakni pengujian parsial untuk masing-masing variabel prediktor dan pengujian asumsi *Cox Proportional Hazard*. Hasil estimasi parameter dan pengujian parsial disajikan pada Tabel 4.11.

Tabel 4.11 Cox PH pada Nasabah *Late Payment*

Variabel	Koefisien	Exp(Koefisien)	S.E. (Koefisien)	Wald	P-value
X_1	0.036	1.037	0.032	1.130	0.259
X_2	0.025	1.025	0.030	0.838	0.402
X_3	-0.060	0.942	0.016	-3.717	0.000
X_4	-0.017	0.983	0.065	-0.255	0.798
X_5	-0.157	0.854	0.073	-2.166	0.030
X_6	-0.200	0.819	0.064	-3.142	0.002

Berdasarkan Tabel 4.11, variabel durasi peminjaman (X_3), jenis nasabah (X_5), dan kepemilikan pinjaman (X_6) berpengaruh signifikan terhadap laju pembayaran untuk nasabah *late payment* di perusahaan *Fintech-X*. Sedangkan variabel lainnya tidak berpengaruh signifikan. Nilai *odds ratio* pada variabel yang signifikan dapat dilihat pada hasil $\exp(koef)$. Misalnya pada variabel kepemilikan pinjaman sebelumnya, diperoleh nilai *odds ratio* sebesar 0.819 yang artinya nasabah yang memiliki pinjaman sebelumnya kemungkinan 0.819 kali lebih lama dalam melakukan pelunasan dibandingkan dengan nasabah yang tidak memiliki pinjaman sebelumnya. Selanjutnya model regresi *Cox Proportional Hazard* dapat dituliskan sebagai berikut.

$$h(t,\boldsymbol{x})_{lp} = h_0(t)\exp(0.036X_1 + 0.025X_2 - 0.060X_3 - 0.017X_4 - 0.157X_5 - 0.200X_6)$$

Tabel 4.12 Pengujian Asumsi Cox PH pada Nasabah *Late Payment*

Variabel	Korelasi	P-Value
X_1	0.017	0.603
X_2	-0.047	0.140
X_3	0.044	0.128
X_4	-0.062	0.045
X_5	0.058	0.066
X_6	0.019	0.544

Namun pemodelan Cox Proportional Hazard memiliki kelemahan yakni asumsi proportional hazard yang harus dipenuhi. Pada Tabel 4.12 dapat dilihat bahwa variabel durasi peminjaman tidak memenuhi asumsi.Karena model *Cox Proportional Hazard* tidak memenuhi asumsi, maka dilakukan pemodelan lain yang tidak memperhatikan asumsi. Pada penelitian ini metode yang digunakan untuk mengatasi permasalahan tersebut adalah dengan *Survival* SVM.

Terdapat 2 parameter γ dan μ serta parameter kernel RBF σ pada model Survival SVM. Pada penelitian ini nilai parameter yang digunakan masing-masing adalah 0.1, 0.3 dan 1. Hasil statistika deskriptif dari prognostik indeks yang dihasilkan adalah sebagai berikut.

Tabel 4.13 Statistika Deskriptif Prognostik Indeks *Survival* SVM Nasabah *Late Payment*

Variabel	Rata-Rata	Median	Standar Deviasi
Prognostik Indeks	25.4831	25.6188	0.9983

Berdasarkan Tabel 4.13 nasabah yang memiliki prognostik indeks di atas rata-rata maka nasabah tersebut masuk dalam kategori *high risk.* Artinya nasabah tersebut memiliki peluang durasi keterlambatan dalam melunasi hutangnya semakin lama. Sedangkan pada nasabah *low risk* merupakan nasabah yang peluang durasi keterlambatan dalam melunasi hutangnya semakin pendek.

Setelah diperoleh nilai prognostik untuk setiap observasi, maka dapat dihitung performa model Survival SVM maupun *Cox Proportional Hazard*. Berdasarkan Tabel 4.14, metode *Survival* SVM jauh lebih baik dari *Cox Proportional Hazard*.

Tabel 4.14 C-Index Masing-Masing Metode untuk Nasabah *Early Payment*

Metode	C-Index
Cox PH	0.4756
Survival SVM	0.6050

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan pembahasan yang telah dilakukan maka dapat diambil kesimpulan sebagai berikut :

1. Hasil yang ditemukan dalam studi simulasi ini menunjukkan bahwa SVM dan *Ensemble* SVM mampu mengkompensasi kinerja regresi logistik dalam beberapa skenario. SVM secara umum memiliki performa sebaik regresi logistik sedangkan *Ensemble* SVM berperforma lebih buruk ketika prediktor kategorik tidak dilibatkan. Penambahan jumlah klaster dan metode klaster yang berbeda dalam *Ensemble* SVM tidak memiliki pengaruh signifikan pada peningkatan peforma klasifikasi. Hal ini karena jumlah klaster yang benar adalah dua.
2. Pemodelan klasifikasi pada data riil dengan menggunakan regresi logistik menunjukkan bahwa variabel yang konsisten signifikan adalah usia X_1 dan jenis nasabah (X_5). Kemudian ketika terjadi perubahan kode pada variabel X_6 ternyata tidak terjadi perubahan tanda pada koefisien. Namun hal tersebut masih ditoleransi karena variabel tersebut tidak signifikan.
3. Perbandingan metode regresi logsitik, SVM, dan *Ensemble* SVM menunjukkan bahwa SVM memiliki peforma yang lebih baik dibandingkan metode lainnya. Walaupun demikian metode terbaik yang dihasilkan masih *out of perform*. Hal tersebut disebabkan rata-rata AUC pada model dengan parameter terbaik masih berada di sekitar 0.5. Sehingga dapat dikatakan variabel dan metode klasifikasi yang digunakan belum dapat mendiskriminasi dengan baik.
4. Pada pemodelan analisis *survival*, model *Survival SVM* memberikan peforma yang lebih baik dibandingkan dengan *Cox Proportional Hazard*. Pada model *Cox Proportional Hazard* diperoleh hasil bahwa variabel durasi peminjaman (X_3) dan jenis nasabah (X_5) berpengaruh signifikan terhadap laju pembayaran untuk nasabah *early payment* di perusahaan *Fintech-X*. Sedangkan pada nasabah *late payment*, variabel durasi

peminjaman (X_3), jenis nasabah (X_5), dan kepemilikan pinjaman (X_6) berpengaruh signifikan terhadap laju pembayaran.

5.2 Saran

Berdasarkan analisis dan pembahasan serta kesimpulan yang didapatkan, terdapat beberapa hal yang dapat menjadi rekomendasi baik untuk penelitian selanjutnya.

1. Untuk studi simulasi lebih lanjut mungkin mengeksplorasi efek data non-linier, *outlier*, variabel prediktor kategori yang saling berkorelasi dan *imbalance class*. Sehingga, dapat diperoleh kesimpulan yang lebih komprehensif.
2. Pada kasus klasifikasi perlu ditambahkan variabel yang relevan agar mampu mendiskriminasi antara nasabah *early payment* dan *late payment* seperti status pernikahan, status kepemilikan rumah, pendapatan, dan lain-lain
3. Pada kasus analisis *survival* perlu dikaji lebih lanjut seberapa besar efek peformansi yang dihasilkan apabila parameter *Survival* SVM di optimasi.

DAFTAR PUSTAKA

Agresti, A., (2013), *Categorical Data Analysis (3rd ed.)*, New Jersey: John Wiley & Sons.

Baesens, B., Gestel, T. V., Stepanova, M., Poel, D. V, & Vanthienen, J. (2005), "Neural Network Survival Analysis for Personal Loan Data", *Journal of the Operational Research Society,* 56, 1089–1098.

Boser, B. E., Guyon, I. M, & Vapnik, V. N., (1992), "A Training Algorithm for Optimal Margin Classifiers". *Proceeding COLT '92 Proceedings of the fifth annual workshop on Computational Learning Theory*, USA, 144-152.

Constangioara, A., (2011), "Consumer Credit Scoring". *Romanian Journal of Economic Forecasting*, 3 162–177.

Cox, D. R., (1972), "Regression models and life-tables (with discussion)". *J. Royal Statist. Society*, Series B, 74 187–220.

Dhilon, I., Guan, Y., & Kulis, B., (2005), "A Unified View of Kernel k-means, Spectral Clustering and Graph Cuts". UTCS Technical Report.

Ghodselahi, A., (2011), "A Hybrid Support Vector Machine Ensemble Model for Credit Scoring". *International Journal of Computer Applications*, 17 (5).

Gu, Q., & Han, J., (2013), "Clustered Support Vector Machines". Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, 307-315.

Gunn, S. R., (1998), *Support Vector Machine for Classification and Regression*: University of Southampton.

Haerdle, W. K., Prastyo, D. D., & Hafner, C. M., (2014), "Support Vector Machines with Evolutionary Model Selection for Default Prediction," in Handbook of Applied Nonparametric and Semiparametric Econometrics and Statistics, J.S. Racine, L. Su, and A. Ullah, Eds. New York: Oxford University Press Inc, 2014, pp. 346-373.

Han, J., Kamber, M., & Pei, J., (2012), *Data Mining : Concepts and Technique 3rd edition.* USA: Morgan Kaufman.

Han, L., Han, L., Zhao, H., (2013), "Orthogonal support vector machine for credit scoring", *Engineering Applications of Artificial Intelligence*, 26, 848–862.

Harrell, F. Jr., Klee, K., Califf, R., Pryor, D., & Rosati, R., (1984), "Regression Modeling Strategies for Improved Prognostic Prediction", *Statistics in Medicine* 1984, Vol. 3, No. 2, 143-152.

Hosmer, D. W., Lemeshow, S., & Sturdivant, X. R., (2013), *Applied Logistic Regression* (3rd ed.), New Jersey: John Wiley & Sons.

Hsu, C., Chang, C., & Lin, C., (2004), *A Practical Guide to Support Vector Classification*, Information Engineering Taiwan University, Taipei.

Johnson, R.A and Winchern, D.W., (2007), *Applied Multivariate Analysis* (Sixth Edition), New Jersey : Prentice Hall Inc.

Kamus Besar Bahasa Indonesia (2017), https://kbbi.web.id/gadai, diakses 10 September 2017.

Khotimah, C., Purnami, S. W., & Prastyo, D. D., (2017), "Additive survival least square support vector machines: A simulation study and its application to cervical cancer prediction". AIP Conference Proceedings, 1902, 050024.

Khotimah, C., Purnami, S. W., & Prastyo, D. D., (2018), "Additive Survival Least Square Support Vector Machines and Feature Selection on Health Data in Indonesia". IEEE Xplore: International Conference on Information and Communications Technology (ICOIACT) 2018.

Kleinbaum, D. G., & Klein, M., (2012), *Survival Analysis: A Self Learning Text* (Third ed.), London: Springer.

Mahjub, H., Faradmal, J., Goli, & S., Soltanian, A., (2016), "Performance Evaluation of Support Vector Regression Models for Survival Analysis: A Simulation Study", (IJACSA) *International Journal of Advanced Computer Science an Application Vol. 7*, No 6.

Mercer, J., (1909), "Foundations of Positive and Negative Type, and Their connection with the Theory of Integral Equations", *Philosophical Transactions of the Royal Society of London*, 25, 3-23.

Moore, D. F., (2016), *Applied Survival Analysis Using R*, London: Springer.

Machin, D., Cheung, Y. B., & Parmar, M. K. B., (2006), *Survival Analysis a Practical Approach,* New Jersey: John Wiley & Sons.

Narain, B., (1992), "Survival Analysis and the Credit Granting Decision". L. C. Thomas, J. N. Crook, D. B. Edelman, eds. *Credit Scoring and Credit Control*. OUP, Oxford, U.K., 109–121.

Prasetyo, E., (2014), *Data Mining Konsep dan Aplikasi Menggunakan MATLAB*, Yogyakarta: Andi.

Robandi, I., & Prasetyo, G. R. A., (2008), *Peramalan Beban Jangka Pendek untuk Hari-Hari Libur dengan Metode Support Vector Machine*, Tugas Akhir, Institut Teknologi Sepuluh Nopember, Surabaya.

Rokach, L., (2009), "Ensemble-based classifiers", *Artificial Intelligence Review*, 33, 1-39.

Santosa, B., (2007). *Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis*, Yogyakarta: Graha Ilmu.

Schoenfeld, D., (1982), "Partial Residual for Proportional Hazard Regression Model", *Biometrika,* 69(1), 239-241.

Scholkopf, B., & Smola, (2002), *Learning with Kernel : Support Vactor Machines, Regulerization, Optimization, and* Beyond, MIT Press, Cambridge.

Stepanova, M. & Thomas, L.C., (2002), "Survival analysis methods for personal loan data". *Journal of the Operational Research Society,* 50(2), 277–289.

Thomas, L. C., J. Banasik, & J. N. Crook., (1999), "Not if but when loans default", *Journal of the Operational Research Society*, 50, 1185–1190.

Van Belle, V. Pleckmans, K., Suykens, J. A., & Van Huffel, S., (2007), "Support Vector Machines for Survival Analysis", *Proceedings of the Third International Conference on Computational Intelligence in Medicine and Healthcare* (CIMED2007), Plymouth (UK), 1-8.

Van Belle, V. Pleckmans, K., Suykens, J. A., & Van Huffel, S., (2008), "Survival SVM: a Practical Scalable Algorithm", *Proceedings of the 16th European Symposium on Artificial Neural Networks* (ESANN2008), Bruges (Belgium), 89-94.

Van Belle, V. Pleckmans, K., Suykens, J. A., & Van Huffel, S., (2010), "Additive Survival Least Square Support Vector Machines", *Statistics in Medicine,* 29(2) : 296-308.

Van Belle, V. Pleckmans, K., Suykens, J. A., & Van Huffel, S., (2011), "Support Vector Methods for Survival Analysis: A Comparison Between Ranking and Regression Approaches", *Artificial Intelligence in Medicine,* 53, 107-118.

Lampiran 1 : Data Nasabah *Early Payment* dan *Late Payment* Gadai Fintech-X

ID	Tanggal Peminjaman	Tanggal Jatuh Tempo	Tanggal Pelunasan	X1	X2	X3	X4	X5	X6	T	Y
17	4/20/2015	8/10/2015	5/25/2015	38	347	4	L	1	0	77	0
18	4/22/2015	5/13/2015	4/30/2015	28	27,500	3	L	1	0	13	0
55	5/5/2015	6/2/2015	7/3/2015	29	1,375	4	P	1	0	31	1
56	5/6/2015	6/3/2015	6/3/2015	44	1,650	4	P	0	1	0	0
60	5/7/2015	5/21/2015	5/29/2015	25	2,365	2	P	0	0	8	1
72	5/7/2015	5/21/2015	6/12/2015	29	2,200	2	L	1	0	22	1
78	5/8/2015	5/22/2015	5/19/2015	22	990	2	P	1	0	3	0
17	5/18/2015	8/17/2015	5/25/2015	38	350	4	L	0	0	84	0
135	5/19/2015	6/9/2015	6/9/2015	24	1,320	3	L	1	0	0	0
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
19097	7/29/2017	8/5/2017	8/1/2017	22	2,222	1	L	0	1	4	0
13918	7/31/2017	8/21/2017	9/4/2017	48	3,172	3	P	0	0	14	1
13918	7/31/2017	8/21/2017	9/4/2017	48	3,234	3	P	0	1	14	1
19329	8/4/2017	8/18/2017	8/18/2017	38	1,096	4	P	0	1	0	0
25076	8/4/2017	8/18/2017	8/30/2017	53	9,099	4	L	0	0	12	1
25076	8/4/2017	8/18/2017	8/30/2017	53	9,275	4	L	0	1	12	1
25143	8/7/2017	9/4/2017	9/2/2017	25	1,265	4	L	0	0	2	0
19571	8/9/2017	8/23/2017	8/23/2017	42	4,682	4	P	0	0	0	0
26656	8/21/2017	9/18/2017	8/31/2017	20	4,576	4	P	1	0	18	0
25343	8/23/2017	9/20/2017	8/23/2017	43	1,602	4	L	1	0	28	0

Keterangan :

X_1 = Usia Nasabah (Tahun)

X_2 = Besaran Peminjaman (Rp dalam ribuan)

X_3 = Durasi Peminjaman (dalam minggu)

X_4 =Jenis Kelamin Nasabah (Laki-laki=1, Perempuan=0)

X_5 =Jenis Nasabah (Nasabah baru=1, lama=0)

X_6 =Kepemilikan Tanggungan(Memiliki pinjaman sebelumnya =1, tidak memiliki=0)

Y = Kategori Nasabah (*Late Payment* =1, *Early Payment* =0)

T = Survival Time (Hari)

Lampiran 2 : Sintaks R Studi Simulasi Skenario 1

```
library(caret)
library(e1071)
library(MLmetrics)
library(SwarmSVM)
library(fmsb)
library(ggpubr)
n=500
iter=100
Eval_glm=matrix(nrow=iter,ncol=4)
Eval_svm=matrix(nrow=iter,ncol=4)
Evalkern_csvm2=matrix(nrow=iter,ncol=4)
Evalkmean_csvm2=matrix(nrow=iter,ncol=4)
Evalkern_csvm3=matrix(nrow=iter,ncol=4)
Evalkmean_csvm3=matrix(nrow=iter,ncol=4)
Evalkern_csvm4=matrix(nrow=iter,ncol=4)
Evalkmean_csvm4=matrix(nrow=iter,ncol=4)
for (i in 1:iter) {
    x1<-rbinom(n,1,0.2)
    x2<-rbinom(n,1,0.2)
    x3<-rbinom(n,1,0.2)
    x4<-rbinom(n,1,0.2)
    x5<- rnorm(n,20,5)
    x6<- rnorm(n,30,5)
    x<-data.frame(x1,x2,x3,x4,x5,x6)
    z <- 2*x$x1+4*x$x2+3*x$x3+0*x$x4+7*x$x5-4*x$x6
    pr <- exp(z) / ( 1 + exp(z))   ## inverse-logit
    y <- rbinom(n, 1, pr)
    df <- data.frame(y=y, x)
    #sum(pr)
    #sum(df$y)
   model_glm <- glm(y~-1+x1+x2+x3+x4+x5+x6,data=df,family = binomial())
   pred_glm <- ifelse(predict(model_glm,df,type="response")>0.5,1,0)
   Eval_glm[i,]<-
c(AUC(pred_glm,df$y),Accuracy(pred_glm,df$y),Sensitivity(pred_glm,df$y,positive="1"),Specificity(pre
d_glm,df$y,positive="1"))
   ModelSVM<-svm(y~-1+x1+x2+x3+x4+x5+x6,data=df,type="C-classification",gamma=0.7,cost=8)
   pred_svm<-predict(ModelSVM, df)
   Eval_svm[i,]<-
c(AUC(pred_svm,df$y),Accuracy(pred_svm,df$y),Sensitivity(pred_svm,df$y,positive="1"),Specificity(pr
ed_svm,df$y,positive="1"))
   #kernel_Kmean_SVM_2
   Modelkern_csvm2 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 2, verbose =
0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
   predkern_csvm2 <- unlist(predict(Modelkern_csvm2, df[,-1]))
   Evalkern_csvm2[i,] <-
c(AUC(predkern_csvm2,df$y),Accuracy(predkern_csvm2,df$y),Sensitivity(predkern_csvm2,df$y,positive
="1"),Specificity(predkern_csvm2,df$y,positive="1"))
   #Kmean_SVM_2
   Modelkmean_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
   predkmean_csvm2 <- unlist(predict(Modelkmean_csvm2, df[,-1]))
   Evalkmean_csvm2[i,] <-
c(AUC(predkmean_csvm2,df$y),Accuracy(predkmean_csvm2,df$y),Sensitivity(predkmean_csvm2,df$y,p
ositive="1"),Specificity(predkmean_csvm2,df$y,positive="1"))
   #kernel_Kmean_SVM_3
   Modelkern_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
   predkern_csvm3 <- unlist(predict(Modelkern_csvm3, df[,-1]))
```

```
   Evalkern_csvm3[i,] <-
c(AUC(predkern_csvm3,df$y),Accuracy(predkern_csvm3,df$y),Sensitivity(predkern_csvm3,df$y,positive
="1"),Specificity(predkern_csvm3,df$y,positive="1"))
   #Kmean_SVM_3
   Modelkmean_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
   predkmean_csvm3 <- unlist(predict(Modelkmean_csvm3, df[,-1]))
   Evalkmean_csvm3[i,] <-
c(AUC(predkmean_csvm3,df$y),Accuracy(predkmean_csvm3,df$y),Sensitivity(predkmean_csvm3,df$y,p
ositive="1"),Specificity(predkmean_csvm3,df$y,positive="1"))
   #kernel_Kmean_SVM_4
   Modelkern_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 4, verbose =
0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
   predkern_csvm4 <- unlist(predict(Modelkern_csvm4, df[,-1]))
   Evalkern_csvm4[i,] <-
c(AUC(predkern_csvm4,df$y),Accuracy(predkern_csvm4,df$y),Sensitivity(predkern_csvm4,df$y,positive
="1"),Specificity(predkern_csvm4,df$y,positive="1"))
   #Kmean_SVM_4
   Modelkmean_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1, centers = 4, verbose = 0,
cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
   predkmean_csvm4 <- unlist(predict(Modelkmean_csvm4, df[,-1]))
   Evalkmean_csvm4[i,] <-
c(AUC(predkmean_csvm4,df$y),Accuracy(predkmean_csvm4,df$y),Sensitivity(predkmean_csvm4,df$y,p
ositive="1"),Specificity(predkmean_csvm4,df$y,positive="1"))
}
Eval_mean=rbind(colMeans(Eval_glm),colMeans(Eval_svm),colMeans(Evalkern_csvm2),colMeans(Eval
kmean_csvm2),colMeans(Evalkern_csvm3),colMeans(Evalkmean_csvm3),colMeans(Evalkern_csvm4),co
lMeans(Evalkmean_csvm4))
Eval_sd=rbind(apply(Eval_glm, 2, sd),apply(Eval_svm, 2, sd),apply(Evalkern_csvm2, 2,
sd),apply(Evalkmean_csvm2, 2, sd),apply(Evalkern_csvm3, 2, sd),apply(Evalkmean_csvm3, 2,
sd),apply(Evalkern_csvm4, 2, sd),apply(Evalkmean_csvm4, 2, sd))
row.names(Eval_mean)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","ker
n_csvm4","kmean_svm4")
colnames(Eval_mean)=c("AUC","Accuracy","Sensitivity","Specificity")
row.names(Eval_sd)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","kern_c
svm4","kmean_svm4")
colnames(Eval_sd)=c("AUC","Accuracy","Sensitivity","Specificity")
Eval_mean
Eval_sd
Trans_Eval=t(Eval_mean)
data_new=as.data.frame(rbind(rep(1,8) , rep(0.95,8) , Trans_Eval))
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9),rgb(0.1,0.1,0.4,0.5) )
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Plot1_1A.jpeg",width=800,height = 600,quality=100)
radarchart( data_new , axistype=1 ,
        #custom polygon
        pcol=colors_border,
        #custom the grid
        cglcol="black", cglty=1, axislabcol="black", seg= 5, caxislabels=seq(0.95,1,0.01),
cglwd=0.4,plwd=3,plty=4,
        #custom labels
        vlcex=1.4)
legend(x=1, y=0.5, legend = rownames(data_new[-c(1,2),]), bty = "n", pch=20,col=colors_border,cex=1,
pt.cex=2.5)
dev.off()

Data_Eval=rbind(Eval_glm,Eval_svm,Evalkern_csvm2,Evalkmean_csvm2,Evalkern_csvm3,Evalkmean_
csvm3,Evalkern_csvm4,Evalkmean_csvm4)
Metode=c(rep("glm",iter),rep("svm",iter),rep("kern_csvm2",iter),rep("kmean_svm2",iter),rep("kern_csvm
3",iter),rep("kmean_svm3",iter),rep("kern_csvm4",iter),rep("kmean_csvm4",iter))
Data_Eval=data.frame(Data_Eval,Metode)
colnames(Data_Eval)=c("AUC","Accuracy","Sensitivity","Specificity","Methods")
```

```
plot1=ggplot(Data_Eval,aes(x=Methods,y=AUC))+geom_boxplot(fill="grey")+theme_minimal()+theme(
axis.text.x = element_text(size=16),panel.border =
element_blank(),panel.grid=element_blank())+xlab("")+ylab("AUC")
plot2=ggplot(Data_Eval,aes(x=Methods,y=Accuracy))+geom_boxplot(fill="grey")+theme_minimal()+the
me(axis.text.x = element_text(size=16),panel.border =
element_blank(),panel.grid=element_blank())+ylab("Accuracy")
plot3=ggplot(Data_Eval,aes(x=Methods,y=Sensitivity))+geom_boxplot(fill="grey")+theme_minimal()+th
eme(axis.text.x = element_text(size=16),panel.border =
element_blank(),panel.grid=element_blank())+xlab("")+ylab("Sensitivity")
plot4=ggplot(Data_Eval,aes(x=Methods,y=Specificity))+geom_boxplot(fill="grey")+theme_minimal()+th
eme(axis.text.x = element_text(size=16),panel.border =
element_blank(),panel.grid=element_blank())+ylab("Specificity")
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Plot2_1A.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot1, plot2 , ncol = 1, nrow = 2)
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Plot3_1A.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot3, plot4 , ncol = 1, nrow = 2)
dev.off()
write.csv(Eval_mean,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Hasil_mean_Sim1A.csv",row.names = FALSE)
write.csv(Eval_sd,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Hasil_sd_Sim1A.csv",row.names = FALSE)
write.csv(Data_Eval,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Hasil_Sim1A.csv",row.names = FALSE)
```

Lampiran 3 : Sintaks R Studi Simulasi Skenario 2

```
library(caret)
library(e1071)
library(MLmetrics)
library(SwarmSVM)
library(fmsb)
library(ggpubr)
library(MASS)

n=500
iter=100
Eval_glm=matrix(nrow=iter,ncol=4)
Eval_svm=matrix(nrow=iter,ncol=4)
Evalkern_csvm2=matrix(nrow=iter,ncol=4)
Evalkmean_csvm2=matrix(nrow=iter,ncol=4)
Evalkern_csvm3=matrix(nrow=iter,ncol=4)
Evalkmean_csvm3=matrix(nrow=iter,ncol=4)
Evalkern_csvm4=matrix(nrow=iter,ncol=4)
Evalkmean_csvm4=matrix(nrow=iter,ncol=4)
for (i in 1:iter) {
 x1<-rbinom(n,1,0.2)
 x2<-rbinom(n,1,0.2)
 x3<-rbinom(n,1,0.2)
 x4<-rbinom(n,1,0.2)
 x5_1<- rnorm(n/2,20,5)
 x5_2<- rnorm(n/2,50,5)
 x5<-c(x5_1,x5_2)
 x6_1<- rnorm(n/2,10,5)
 x6_2<- rnorm(n/2,30,5)
 x6<-c(x6_1,x6_2)
 x<-data.frame(x1,x2,x3,x4,x5,x6)
 z <- 2*x$x1+4*x$x2+3*x$x3+0*x$x4+4*x$x5-9*x$x6
 pr <- exp(z) / ( 1 + exp(z))   ## inverse-logit
 y <- rbinom(n, 1, pr)
 df <- data.frame(y=y, x)

 model_glm <- glm(y~-1+x1+x2+x3+x4+x5+x6,data=df,family = binomial())
 pred_glm <- ifelse(predict(model_glm,df,type="response")>0.5,1,0)
 Eval_glm[i,]<-
        c(AUC(pred_glm,df$y),Accuracy(pred_glm,df$y),Sensitivity(pred_glm,df$y,positive="1"),
        Specificity(pred_glm,df$y,positive="1"))
 ModelSVM<-svm(y~-1+x1+x2+x3+x4+x5+x6,data=df,type="C-classification",gamma=0.7,cost=8)
 pred_svm<-predict(ModelSVM, df)
 Eval_svm[i,]<-
        c(AUC(pred_svm,df$y),Accuracy(pred_svm,df$y),Sensitivity(pred_svm,df$y,positive="1"),
        Specificity(pred_svm,df$y,positive="1"))

 #kernel_Kmean_SVM_2
 Modelkern_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm2 <- unlist(predict(Modelkern_csvm2, df[,-1]))
 Evalkern_csvm2[i,] <- c(AUC(predkern_csvm2,df$y),Accuracy(predkern_csvm2,df$y),
        Sensitivity(predkern_csvm2,df$y,positive="1"),Specificity(predkern_csvm2,df$y,positive="1"))

 #Kmean_SVM_2
 Modelkmean_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
        0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
 predkmean_csvm2 <- unlist(predict(Modelkmean_csvm2, df[,-1]))
 Evalkmean_csvm2[i,] <- c(AUC(predkmean_csvm2,df$y),Accuracy(predkmean_csvm2,df$y)
        ,Sensitivity(predkmean_csvm2,df$y,positive="1"),Specificity(predkmean_csvm2,df$y,positive=
        "1"))
 #kernel_Kmean_SVM_3
```

```
  Modelkern_csvm3 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 3, verbose =
          0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm3 <- unlist(predict(Modelkern_csvm3, df[,-1]))
  Evalkern_csvm3[i,] <- c(AUC(predkern_csvm3,df$y),Accuracy(predkern_csvm3,df$y)
          ,Sensitivity(predkern_csvm3,df$y,positive="1"),Specificity(predkern_csvm3,df$y,positive="1"))

  #Kmean_SVM_3
  Modelkmean_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
          0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm3 <- unlist(predict(Modelkmean_csvm3, df[,-1]))
  Evalkmean_csvm3[i,] <- c(AUC(predkmean_csvm3,df$y),Accuracy(predkmean_csvm3,df$y)
          ,Sensitivity(predkmean_csvm3,df$y,positive="1"),Specificity(predkmean_csvm3,df$y,positive=
          "1"))

  #kernel_Kmean_SVM_4
  Modelkern_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 4, verbose =
          0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm4 <- unlist(predict(Modelkern_csvm4, df[,-1]))
  Evalkern_csvm4[i,] <- c(AUC(predkern_csvm4,df$y),Accuracy(predkern_csvm4,df$y)
          ,Sensitivity(predkern_csvm4,df$y,positive="1"),Specificity(predkern_csvm4,df$y,positive="1"))

  #Kmean_SVM_4
  Modelkmean_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1, centers = 4, verbose = 0,
          cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm4 <- unlist(predict(Modelkmean_csvm4, df[,-1]))
  Evalkmean_csvm4[i,] <-c(AUC(predkmean_csvm4,df$y),Accuracy(predkmean_csvm4,df$y),
          Sensitivity(predkmean_csvm4,df$y,positive="1"),Specificity(predkmean_csvm4,df$y,positive="
          1"))
}

Eval_mean=rbind(colMeans(Eval_glm),colMeans(Eval_svm),colMeans(Evalkern_csvm2),colMeans(Eval
          kmean_csvm2),colMeans(Evalkern_csvm3),colMeans(Evalkmean_csvm3),colMeans(Evalkern_
          csvm4),colMeans(Evalkmean_csvm4))
Eval_sd=rbind(apply(Eval_glm, 2, sd),apply(Eval_svm, 2, sd),apply(Evalkern_csvm2, 2,
          sd),apply(Evalkmean_csvm2, 2, sd),apply(Evalkern_csvm3, 2, sd),apply(Evalkmean_csvm3, 2,
          sd),apply(Evalkern_csvm4, 2, sd),apply(Evalkmean_csvm4, 2, sd))
row.names(Eval_mean)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","ker
          n_csvm4","kmean_svm4")
colnames(Eval_mean)=c("AUC","Accuracy","Sensitivity","Specificity")
row.names(Eval_sd)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","kern_c
          svm4","kmean_svm4")
colnames(Eval_sd)=c("AUC","Accuracy","Sensitivity","Specificity")
Eval_mean
Eval_sd
Trans_Eval=t(Eval_mean)
data_new=as.data.frame(rbind(rep(1,8) , rep(0.95,8) , Trans_Eval))
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9),rgb(0.1,0.1,0.4,0.5) )
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
          Simulasi/Plot1_2A.jpeg",width=800,height = 600,quality=100)
radarchart( data_new , axistype=1 ,
        #custom polygon
        pcol=colors_border,
        #custom the grid
        cglcol="black", cglty=1, axislabcol="black", seg= 5, caxislabels=seq(0.95,1,0.01),
          cglwd=0.4,plwd=3,plty=4,
        #custom labels
        vlcex=1.4)
legend(x=1, y=0.5, legend = rownames(data_new[-c(1,2),]), bty = "n", pch=20,col=colors_border,cex=1,
          pt.cex=2.5)
dev.off()

Data_Eval=rbind(Eval_glm,Eval_svm,Evalkern_csvm2,Evalkmean_csvm2,Evalkern_csvm3,Evalkmean_
          csvm3,Evalkern_csvm4,Evalkmean_csvm4)
```

```
Metode=c(rep("glm",iter),rep("svm",iter),rep("kern_csvm2",iter),rep("kmean_svm2",iter),rep("kern_csvm
        3",iter),rep("kmean_svm3",iter),rep("kern_csvm4",iter),rep("kmean_csvm4",iter))
Data_Eval=data.frame(Data_Eval,Metode)
colnames(Data_Eval)=c("AUC","Accuracy","Sensitivity","Specificity","Methods")
plot1=ggplot(Data_Eval,aes(x=Methods,y=AUC))+geom_boxplot(fill="grey")+theme_minimal()+theme(
        axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("") +ylab("AUC")
plot2=ggplot(Data_Eval,aes(x=Methods,y=Accuracy))+geom_boxplot(fill="grey")+theme_minimal()+the
        me(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Accuracy")
plot3=ggplot(Data_Eval,aes(x=Methods,y=Sensitivity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("")+ylab("Specificity")
plot4=ggplot(Data_Eval,aes(x=Methods,y=Specificity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Sensitivity")
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot2_2A.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot1, plot2 , ncol = 1, nrow = 2)
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot3_2A.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot3, plot4 , ncol = 1, nrow = 2)
dev.off()
write.csv(Eval_mean,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_mean_Sim2A.csv",row.names = FALSE)
write.csv(Eval_sd,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_sd_Sim2A.csv",row.names = FALSE)
write.csv(Data_Eval,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_Sim2A.csv",row.names = FALSE)
```

Lampiran 4 : Sintaks R Studi Simulasi Skenario 3

```
library(caret)
library(e1071)
library(MLmetrics)
library(SwarmSVM)
library(fmsb)
library(ggpubr)
library(MASS)
n=500
iter=100
Eval_glm=matrix(nrow=iter,ncol=4)
Eval_svm=matrix(nrow=iter,ncol=4)
Evalkern_csvm2=matrix(nrow=iter,ncol=4)
Evalkmean_csvm2=matrix(nrow=iter,ncol=4)
Evalkern_csvm3=matrix(nrow=iter,ncol=4)
Evalkmean_csvm3=matrix(nrow=iter,ncol=4)
Evalkern_csvm4=matrix(nrow=iter,ncol=4)
Evalkmean_csvm4=matrix(nrow=iter,ncol=4)
mu_1=c(20,40)
mu_2=c(50,70)
sigma_1=matrix(c(25,16,16,25),2,2)
sigma_2=matrix(c(25,16,16,25),2,2)
for (i in 1:iter) {
 x1<-rbinom(n,1,0.2)
 x2<-rbinom(n,1,0.2)
 x3<-rbinom(n,1,0.2)
 x4<-rbinom(n,1,0.2)
 x5_1<- rnorm(n/2,20,5)
 x5_2<- rnorm(n/2,50,5)
 x5<-c(x5_1,x5_2)
 x6_1<- rnorm(n/2,10,5)
 x6_2<- rnorm(n/2,30,5)
 x6<-c(x6_1,x6_2)
 x7_x8_1 <- mvrnorm(n/2, mu_1, sigma_1, empirical = TRUE)
 colnames(x7_x8_1)=c("x7","x8")
 x7_x8_2 <- mvrnorm(n/2, mu_2, sigma_2, empirical = TRUE)
 colnames(x7_x8_2)=c("x7","x8")
 x7_x8<-rbind(x7_x8_1,x7_x8_2)
 x7<-x7_x8[,1]
 x8<-x7_x8[,2]
 x<-data.frame(x1,x2,x3,x4,x5,x6,x7,x8)
 z <- 2*x$x1+4*x$x2+3*x$x3+0*x$x4+4*x$x5-4*x$x6+8*x7-6*x8
 pr <- exp(z) / ( 1 + exp(z))  ## inverse-logit
 y <- rbinom(n, 1, pr)
 df <- data.frame(y=y, x)
 model_glm <- glm(y~-1+x1+x2+x3+x4+x5+x6+x7+x8,data=df,family = binomial())
 pred_glm <- ifelse(predict(model_glm,df,type="response")>0.5,1,0)
 Eval_glm[i,]<-
       c(AUC(pred_glm,df$y),Accuracy(pred_glm,df$y),Sensitivity(pred_glm,df$y,positive="1")
       ,Specificity(pred_glm,df$y,positive="1"))
 ModelSVM<-svm(y~-1+x1+x2+x3+x4+x5+x6+x7+x8,data=df,type="C-
       classification",gamma=0.7,cost=8)
 pred_svm<-predict(ModelSVM, df)
 Eval_svm[i,]<-
       c(AUC(pred_svm,df$y),Accuracy(pred_svm,df$y),Sensitivity(pred_svm,df$y,positive="1")
       ,Specificity(pred_svm,df$y,positive="1"))

 #kernel_Kmean_SVM_2
 Modelkern_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
       0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm2 <- unlist(predict(Modelkern_csvm2, df[,-1]))
 Evalkern_csvm2[i,] <- c(AUC(predkern_csvm2,df$y),Accuracy(predkern_csvm2,df$y),
       Sensitivity(predkern_csvm2,df$y,positive="1"),Specificity(predkern_csvm2,df$y,positive="1"))
```

```
 #Kmean_SVM_2
 Modelkmean_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
       0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
 predkmean_csvm2 <- unlist(predict(Modelkmean_csvm2, df[,-1]))
 Evalkmean_csvm2[i,] <- c(AUC(predkmean_csvm2,df$y),Accuracy(predkmean_csvm2,df$y),
       Sensitivity(predkmean_csvm2,df$y,positive="1"),Specificity(predkmean_csvm2,df$y,positive="1
       "))

 #kernel_Kmean_SVM_3
 Modelkern_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
       0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm3 <- unlist(predict(Modelkern_csvm3, df[,-1]))
 Evalkern_csvm3[i,] <- c(AUC(predkern_csvm3,df$y),Accuracy(predkern_csvm3,df$y),
       Sensitivity(predkern_csvm3,df$y,positive="1"),Specificity(predkern_csvm3,df$y,positive="1"))

 #Kmean_SVM_3
 Modelkmean_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
       0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
 predkmean_csvm3 <- unlist(predict(Modelkmean_csvm3, df[,-1]))
 Evalkmean_csvm3[i,] <- c(AUC(predkmean_csvm3,df$y),Accuracy(predkmean_csvm3,df$y)
       ,Sensitivity(predkmean_csvm3,df$y,positive="1"),Specificity(predkmean_csvm3,df$y,positive="1
       "))

 #kernel_Kmean_SVM_4
 Modelkern_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 4, verbose =
       0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm4 <- unlist(predict(Modelkern_csvm4, df[,-1]))
 Evalkern_csvm4[i,] <- c(AUC(predkern_csvm4,df$y),Accuracy(predkern_csvm4,df$y)
       ,Sensitivity(predkern_csvm4,df$y,positive="1"),Specificity(predkern_csvm4,df$y,positive="1"))

 #Kmean_SVM_4
 Modelkmean_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1, centers = 4, verbose = 0,
       cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
 predkmean_csvm4 <- unlist(predict(Modelkmean_csvm4, df[,-1]))
 Evalkmean_csvm4[i,] <-c(AUC(predkmean_csvm4,df$y),Accuracy(predkmean_csvm4,df$y),
       Sensitivity(predkmean_csvm4,df$y,positive="1"),Specificity(predkmean_csvm4,df$y,positive="1
       "))
}
Eval_mean=rbind(colMeans(Eval_glm),colMeans(Eval_svm),colMeans(Evalkern_csvm2),colMeans(Eval
       kmean_csvm2),colMeans(Evalkern_csvm3),colMeans(Evalkmean_csvm3),colMeans(Evalkern_cs
       vm4),colMeans(Evalkmean_csvm4))
Eval_sd=rbind(apply(Eval_glm, 2, sd),apply(Eval_svm, 2, sd),apply(Evalkern_csvm2, 2,
       sd),apply(Evalkmean_csvm2, 2, sd),apply(Evalkern_csvm3, 2, sd),apply(Evalkmean_csvm3, 2,
       sd),apply(Evalkern_csvm4, 2, sd),apply(Evalkmean_csvm4, 2, sd))
row.names(Eval_mean)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","ker
       n_csvm4","kmean_svm4")
colnames(Eval_mean)=c("AUC","Accuracy","Sensitivity","Specificity")
row.names(Eval_sd)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","kern_c
       svm4","kmean_svm4")
colnames(Eval_sd)=c("AUC","Accuracy","Sensitivity","Specificity")
Eval_mean
Eval_sd
Trans_Eval=t(Eval_mean)
data_new=as.data.frame(rbind(rep(1,8) , rep(0.95,8) , Trans_Eval))
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9),rgb(0.1,0.1,0.4,0.5) )
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
       Simulasi/Plot1_3A.jpeg",width=800,height = 600,quality=100)
radarchart( data_new , axistype=1 ,
       #custom polygon
       pcol=colors_border,
       #custom the grid
```

```
        cglcol="black", cglty=1, axislabcol="black", seg= 5, caxislabels=seq(0.95,1,0.01),
        cglwd=0.4,plwd=3,plty=4,
        #custom labels
        vlcex=1.4)
legend(x=1, y=0.5, legend = rownames(data_new[-c(1,2),]), bty = "n", pch=20,col=colors_border,cex=1,
        pt.cex=2.5)
dev.off()

Data_Eval=rbind(Eval_glm,Eval_svm,Evalkern_csvm2,Evalkmean_csvm2,Evalkern_csvm3,Evalkmean_
        csvm3,Evalkern_csvm4,Evalkmean_csvm4)
Metode=c(rep("glm",iter),rep("svm",iter),rep("kern_csvm2",iter),rep("kmean_svm2",iter),rep("kern_csvm
        3",iter),rep("kmean_svm3",iter),rep("kern_csvm4",iter),rep("kmean_csvm4",iter))
Data_Eval=data.frame(Data_Eval,Metode)
colnames(Data_Eval)=c("AUC","Accuracy","Sensitivity","Specificity","Methods")
plot1=ggplot(Data_Eval,aes(x=Methods,y=AUC))+geom_boxplot(fill="grey")+theme_minimal()+theme(
        axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("") +ylab("AUC")
plot2=ggplot(Data_Eval,aes(x=Methods,y=Accuracy))+geom_boxplot(fill="grey")+theme_minimal()+the
        me(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Accuracy")
plot3=ggplot(Data_Eval,aes(x=Methods,y=Sensitivity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("")+ylab("Specificity")
plot4=ggplot(Data_Eval,aes(x=Methods,y=Specificity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Sensitivity")
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot2_3A.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot1, plot2 , ncol = 1, nrow = 2)
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot3_3A.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot3, plot4 , ncol = 1, nrow = 2)
dev.off()
write.csv(Eval_mean,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_mean_Sim3A.csv",row.names = FALSE)
write.csv(Eval_sd,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_sd_Sim3A.csv",row.names = FALSE)
write.csv(Data_Eval,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_Sim3A.csv",row.names = FALSE)
```

Lampiran 5 : Sintaks R Studi Simulasi Skenario 4

```
library(caret)
library(e1071)
library(MLmetrics)
library(SwarmSVM)
library(fmsb)

n=500
iter=100
Eval_glm=matrix(nrow=iter,ncol=4)
Eval_svm=matrix(nrow=iter,ncol=4)
Evalkern_csvm2=matrix(nrow=iter,ncol=4)
Evalkmean_csvm2=matrix(nrow=iter,ncol=4)
Evalkern_csvm3=matrix(nrow=iter,ncol=4)
Evalkmean_csvm3=matrix(nrow=iter,ncol=4)
Evalkern_csvm4=matrix(nrow=iter,ncol=4)
Evalkmean_csvm4=matrix(nrow=iter,ncol=4)
for (i in 1:iter) {
 x1<-rbinom(n,1,0.2)
 x2<-rbinom(n,1,0.2)
 x3<-rbinom(n,1,0.2)
 x4<-rbinom(n,1,0.2)
 x5_1<- rnorm(n/2,20,5)
 x5_2<- rnorm(n/2,50,5)
 x5<-c(x5_1,x5_2)
 x6_1<- rnorm(n/2,10,5)
 x6_2<- rnorm(n/2,30,5)
 x6<-c(x6_1,x6_2)
 x7<-5*x5-3*x6
 x<-data.frame(x1,x2,x3,x4,x5,x6,x7)
 z <- 2*x$x1+4*x$x2+3*x$x3+0*x$x4+13*x$x5-3*x$x6-3.5*x$x7
 pr <- exp(z) / ( 1 + exp(z))   ## inverse-logit
 y <- rbinom(n, 1, pr)
 df <- data.frame(y=y, x)
 #sum(pr)
 #sum(df$y)
 model_glm <- glm(y~-1+x1+x2+x3+x4+x5+x6+x7,data=df,family=binomial())
 pred_glm <- ifelse(predict(model_glm,df,type="response")>0.5,1,0)
 Eval_glm[i,]<-
        c(AUC(pred_glm,df$y),Accuracy(pred_glm,df$y),Sensitivity(pred_glm,df$y,positive="1"),
        Specificity(pred_glm,df$y,positive="1"))
 ModelSVM<-svm(y~-1+x1+x2+x3+x4+x5+x6+x7,data=df,type="C-classification",gamma=0.7,cost=8)
 pred_svm<-predict(ModelSVM, df)
 Eval_svm[i,]<-
        c(AUC(pred_svm,df$y),Accuracy(pred_svm,df$y),Sensitivity(pred_svm,df$y,positive="1"),
        Specificity(pred_svm,df$y,positive="1"))

 #kernel_Kmean_SVM_2
 Modelkern_csvm2 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 2, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm2 <- unlist(predict(Modelkern_csvm2, df[,-1]))
 Evalkern_csvm2[i,]<-c(AUC(predkern_csvm2,df$y),Accuracy(predkern_csvm2,df$y),
        Sensitivity(predkern_csvm2,df$y,positive="1"),Specificity(predkern_csvm2,df$y,positive="1"))

 #Kmean_SVM_2
 Modelkmean_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
        0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
 predkmean_csvm2 <- unlist(predict(Modelkmean_csvm2, df[,-1]))
 Evalkmean_csvm2[i,]<-c(AUC(predkmean_csvm2,df$y),Accuracy(predkmean_csvm2,df$y),
        Sensitivity(predkmean_csvm2,df$y,positive="1"),Specificity(predkmean_csvm2,df$y,positive="
        1"))
```

```
  #kernel_Kmean_SVM_3
  Modelkern_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm3 <- unlist(predict(Modelkern_csvm3, df[,-1]))
  Evalkern_csvm3[i,]<-c(AUC(predkern_csvm3,df$y),Accuracy(predkern_csvm3,df$y),
        Sensitivity(predkern_csvm3,df$y,positive="1"),Specificity(predkern_csvm3,df$y,positive="1"))

  #Kmean_SVM_3
  Modelkmean_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
        0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm3 <- unlist(predict(Modelkmean_csvm3, df[,-1]))
  Evalkmean_csvm3[i,]<-c(AUC(predkmean_csvm3,df$y),Accuracy(predkmean_csvm3,df$y),
        Sensitivity(predkmean_csvm3,df$y,positive="1"),Specificity(predkmean_csvm3,df$y,positive="
        1"))

  #kernel_Kmean_SVM_4
  Modelkern_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 4, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm4 <- unlist(predict(Modelkern_csvm4, df[,-1]))
  Evalkern_csvm4[i,]<-c(AUC(predkern_csvm4,df$y),Accuracy(predkern_csvm4,df$y),
        Sensitivity(predkern_csvm4,df$y,positive="1"),Specificity(predkern_csvm4,df$y,positive="1"))

  #Kmean_SVM_4
  Modelkmean_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1, centers = 4, verbose = 0,
        cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm4 <- unlist(predict(Modelkmean_csvm4, df[,-1]))
  Evalkmean_csvm4[i,]<-c(AUC(predkmean_csvm4,df$y),Accuracy(predkmean_csvm4,df$y),
        Sensitivity(predkmean_csvm4,df$y,positive="1"),Specificity(predkmean_csvm4,df$y,positive="
        1"))
}
Eval_mean=rbind(colMeans(Eval_glm),colMeans(Eval_svm),colMeans(Evalkern_csvm2),colMeans(Ev
        alkmean_csvm2),colMeans(Evalkern_csvm3),colMeans(Evalkmean_csvm3),colMeans(Evalkern
        _csvm4),colMeans(Evalkmean_csvm4))
Eval_sd=rbind(apply(Eval_glm,2,sd),apply(Eval_svm,2,sd),apply(Evalkern_csvm2,2,sd),
        apply(Evalkmean_csvm2,2,sd),apply(Evalkern_csvm3,2,sd),apply(Evalkmean_csvm3,2,sd),
        apply(Evalkern_csvm4, 2, sd),apply(Evalkmean_csvm4, 2, sd))
row.names(Eval_mean)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","k
        ern_csvm4","kmean_svm4")
colnames(Eval_mean)=c("AUC","Accuracy","Sensitivity","Specificity")
row.names(Eval_sd)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","kern
        _csvm4","kmean_svm4")
colnames(Eval_sd)=c("AUC","Accuracy","Sensitivity","Specificity")
Eval_mean
Eval_sd
Trans_Eval=t(Eval_mean)
data_new=as.data.frame(rbind(rep(1,8) , rep(0.95,8) , Trans_Eval))
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9),rgb(0.1,0.1,0.4,0.5) )
jpeg(file="D:/Perkuliahan/S2                                    Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot1_4A.jpeg",width=800,height = 600,quality=100)
radarchart( data_new , axistype=1 ,
        #custom polygon
        pcol=colors_border,
        #custom the grid
        cglcol="black",cglty=1,axislabcol="black",seg=5,caxislabels=seq(0.95,1,0.01),
         cglwd=0.4,plwd=3,plty=4,
        #custom labels
        vlcex=1.4)
legend(x=1, y=0.5, legend = rownames(data_new[-c(1,2),]), bty = "n", pch=20,col=colors_border,cex=1,
        pt.cex=2.5)
dev.off()
```

```
Data_Eval=rbind(Eval_glm,Eval_svm,Evalkern_csvm2,Evalkmean_csvm2,Evalkern_csvm3,Evalkmean
    _csvm3,Evalkern_csvm4,Evalkmean_csvm4)
Metode=c(rep("glm",iter),rep("svm",iter),rep("kern_csvm2",iter),rep("kmean_svm2",iter),rep("kern_csv
    m3",iter),rep("kmean_svm3",iter),rep("kern_csvm4",iter),rep("kmean_csvm4",iter))
Data_Eval=data.frame(Data_Eval,Metode)
colnames(Data_Eval)=c("AUC","Accuracy","Sensitivity","Specificity","Methods")
plot1=ggplot(Data_Eval,aes(x=Methods,y=AUC))+geom_boxplot(fill="grey")+theme_minimal()+theme
    (axis.text.x = element_text(size=16),panel.border =
    element_blank(),panel.grid=element_blank())+xlab("") +ylab("AUC")
plot2=ggplot(Data_Eval,aes(x=Methods,y=Accuracy))+geom_boxplot(fill="grey")+theme_minimal()+th
    eme(axis.text.x=element_text(size=16),panel.border=element_blank(),panel.grid=element_blank
    ()) +ylab("Accuracy")
plot3=ggplot(Data_Eval,aes(x=Methods,y=Sensitivity))+geom_boxplot(fill="grey")+theme_minimal()+t
    heme(axis.text.x=element_text(size=16),panel.border=element_blank(),panel.grid=element_blan
    k())+xlab("") +ylab("Specificity")
plot4=ggplot(Data_Eval,aes(x=Methods,y=Specificity))+geom_boxplot(fill="grey")+theme_minimal()+t
    heme(axis.text.x=element_text(size=16),panel.border=element_blank(),panel.grid=element_blan
    k()) +ylab("Sensitivity")
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
    Simulasi/Plot2_4A.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot1, plot2 , ncol = 1, nrow = 2)
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
    Simulasi/Plot3_4A.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot3, plot4 , ncol = 1, nrow = 2)
dev.off()
write.csv(Eval_mean,"D:/Perkuliahan/S2Statistika/Thesis/PROPOSAL/Hasil
    Simulasi/Hasil_mean_Sim4A.csv",row.names = FALSE)
write.csv(Eval_sd,"D:/Perkuliahan/S2Statistika/Thesis/PROPOSAL/Hasil
    Simulasi/Hasil_sd_Sim4A.csv",row.names = FALSE)
write.csv(Data_Eval,"D:/Perkuliahan/S2Statistika/Thesis/PROPOSAL/Hasil
    Simulasi/Hasil_Sim4A.csv",row.names = FALSE)
```

Lampiran 6 : Sintaks R Studi Simulasi Skenario 5

```
library(caret)
library(e1071)
library(MLmetrics)
library(SwarmSVM)

n=500
iter=100
Eval_glm=matrix(nrow=iter,ncol=4)
Eval_svm=matrix(nrow=iter,ncol=4)
Evalkern_csvm2=matrix(nrow=iter,ncol=4)
Evalkmean_csvm2=matrix(nrow=iter,ncol=4)
Evalkern_csvm3=matrix(nrow=iter,ncol=4)
Evalkmean_csvm3=matrix(nrow=iter,ncol=4)
Evalkern_csvm4=matrix(nrow=iter,ncol=4)
Evalkmean_csvm4=matrix(nrow=iter,ncol=4)
mu_1=c(20,40)
mu_2=c(10,10)
sigma_1=matrix(c(25,16,16,25),2,2)
sigma_2=matrix(c(25,16,16,25),2,2)
for (i in 1:iter) {
  x1<-rbinom(n,1,0.2)
  x2<-rbinom(n,1,0.2)
  x3<-rbinom(n,1,0.2)
  x4<-rbinom(n,1,0.2)
  x5_1<- rnorm(n/2,20,5)
  x5_2<- rnorm(n/2,50,5)
  x5<-c(x5_1,x5_2)
  x6_1<- rnorm(n/2,10,5)
  x6_2<- rnorm(n/2,30,5)
  x6<-c(x6_1,x6_2)
  x7<-5*x5-3*x6
  x8_x9_1 <- mvrnorm(n/2, mu_1, sigma_1, empirical = TRUE)
  colnames(x8_x9_1)=c("x8","x9")
  x8_x9_2 <- mvrnorm(n/2, mu_2, sigma_2, empirical = TRUE)
  colnames(x8_x9_2)=c("x8","x9")
  x8_x9<-rbind(x8_x9_1,x8_x9_2)
  x8<-x8_x9[,1]
  x9<-x8_x9[,2]
  x<-data.frame(x1,x2,x3,x4,x5,x6,x7,x8,x9)
  z <- 2*x$x1+4*x$x2+3*x$x3+0*x$x4+13*x$x5-3*x$x6-3.5*x$x7+5.5*x$x8-3*x$x9
  pr <- exp(z) / ( 1 + exp(z))   ## inverse-logit
  y <- rbinom(n, 1, pr)
  df <- data.frame(y=y, x)
  #sum(pr)
  #sum(df$y)
  model_glm <- glm(y~-1+x1+x2+x3+x4+x5+x6+x7+x8+x9,data=df,family=binomial())
  pred_glm <- ifelse(predict(model_glm,df,type="response")>0.5,1,0)
  Eval_glm[i,]<-
        c(AUC(pred_glm,df$y),Accuracy(pred_glm,df$y),Sensitivity(pred_glm,df$y,positive="1"),Specifi
        city(pred_glm,df$y,positive="1"))
  ModelSVM<-svm(y~-1+x1+x2+x3+x4+x5+x6+x7+x8+x9,data=df,type="C-
        classification",gamma=0.7,cost=8)
  pred_svm<-predict(ModelSVM, df)
  Eval_svm[i,]<-
        c(AUC(pred_svm,df$y),Accuracy(pred_svm,df$y),Sensitivity(pred_svm,df$y,positive="1"),Specif
        icity(pred_svm,df$y,positive="1"))
  #kernel_Kmean_SVM_2
  Modelkern_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm2 <- unlist(predict(Modelkern_csvm2, df[,-1]))
```

```
  Evalkern_csvm2[i,] <-
        c(AUC(predkern_csvm2,df$y),Accuracy(predkern_csvm2,df$y),Sensitivity(predkern_csvm2,df$y,
        positive="1"),Specificity(predkern_csvm2,df$y,positive="1"))
  #Kmean_SVM_2
  Modelkmean_csvm2 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 2, verbose =
        0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm2 <- unlist(predict(Modelkmean_csvm2, df[,-1]))
  Evalkmean_csvm2[i,] <-
        c(AUC(predkmean_csvm2,df$y),Accuracy(predkmean_csvm2,df$y),Sensitivity(predkmean_csvm
        2,df$y,positive="1"),Specificity(predkmean_csvm2,df$y,positive="1"))
  #kernel_Kmean_SVM_3
  Modelkern_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm3 <- unlist(predict(Modelkern_csvm3, df[,-1]))
  Evalkern_csvm3[i,] <-
        c(AUC(predkern_csvm3,df$y),Accuracy(predkern_csvm3,df$y),Sensitivity(predkern_csvm3,df$y,
        positive="1"),Specificity(predkern_csvm3,df$y,positive="1"))
  #Kmean_SVM_3
  Modelkmean_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
        0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm3 <- unlist(predict(Modelkmean_csvm3, df[,-1]))
  Evalkmean_csvm3[i,] <-
        c(AUC(predkmean_csvm3,df$y),Accuracy(predkmean_csvm3,df$y),Sensitivity(predkmean_csvm
        3,df$y,positive="1"),Specificity(predkmean_csvm3,df$y,positive="1"))
  #kernel_Kmean_SVM_4
  Modelkern_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 4, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm4 <- unlist(predict(Modelkern_csvm4, df[,-1]))
  Evalkern_csvm4[i,] <-
        c(AUC(predkern_csvm4,df$y),Accuracy(predkern_csvm4,df$y),Sensitivity(predkern_csvm4,df$y,
        positive="1"),Specificity(predkern_csvm4,df$y,positive="1"))
  #Kmean_SVM_4
  Modelkmean_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1, centers = 4, verbose = 0,
        cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm4 <- unlist(predict(Modelkmean_csvm4, df[,-1]))
  Evalkmean_csvm4[i,] <-
        c(AUC(predkmean_csvm4,df$y),Accuracy(predkmean_csvm4,df$y),Sensitivity(predkmean_csvm
        4,df$y,positive="1"),Specificity(predkmean_csvm4,df$y,positive="1"))
}
Eval_mean=rbind(colMeans(Eval_glm),colMeans(Eval_svm),colMeans(Evalkern_csvm2),colMeans(Eval
        kmean_csvm2),colMeans(Evalkern_csvm3),colMeans(Evalkmean_csvm3),colMeans(Evalkern_cs
        vm4),colMeans(Evalkmean_csvm4))
Eval_sd=rbind(apply(Eval_glm, 2, sd),apply(Eval_svm, 2, sd),apply(Evalkern_csvm2, 2,
        sd),apply(Evalkmean_csvm2, 2, sd),apply(Evalkern_csvm3, 2, sd),apply(Evalkmean_csvm3, 2,
        sd),apply(Evalkern_csvm4, 2, sd),apply(Evalkmean_csvm4, 2, sd))
row.names(Eval_mean)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","ker
        n_csvm4","kmean_svm4")
colnames(Eval_mean)=c("AUC","Accuracy","Sensitivity","Specificity")
row.names(Eval_sd)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","kern_c
        svm4","kmean_svm4")
colnames(Eval_sd)=c("AUC","Accuracy","Sensitivity","Specificity")
Eval_mean
Eval_sd
Trans_Eval=t(Eval_mean)
data_new=as.data.frame(rbind(rep(1,8) , rep(0.95,8) , Trans_Eval))
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9),rgb(0.1,0.1,0.4,0.5) )
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot1_5A.jpeg",width=800,height = 600,quality=100)
radarchart( data_new , axistype=1 ,
        #custom polygon
        pcol=colors_border,
        #custom the grid
```

```
        cglcol="black", cglty=1, axislabcol="black", seg= 5, caxislabels=seq(0.95,1,0.01),
        cglwd=0.4,plwd=3,plty=4,
        #custom labels
        vlcex=1.4)
legend(x=1, y=0.5, legend = rownames(data_new[-c(1,2),]), bty = "n", pch=20,col=colors_border,cex=1,
        pt.cex=2.5)
dev.off()

Data_Eval=rbind(Eval_glm,Eval_svm,Evalkern_csvm2,Evalkmean_csvm2,Evalkern_csvm3,Evalkmean_
        csvm3,Evalkern_csvm4,Evalkmean_csvm4)
Metode=c(rep("glm",iter),rep("svm",iter),rep("kern_csvm2",iter),rep("kmean_svm2",iter),rep("kern_csvm
        3",iter),rep("kmean_svm3",iter),rep("kern_csvm4",iter),rep("kmean_csvm4",iter))
Data_Eval=data.frame(Data_Eval,Metode)
colnames(Data_Eval)=c("AUC","Accuracy","Sensitivity","Specificity","Methods")
plot1=ggplot(Data_Eval,aes(x=Methods,y=AUC))+geom_boxplot(fill="grey")+theme_minimal()+theme(
        axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("") +ylab("AUC")
plot2=ggplot(Data_Eval,aes(x=Methods,y=Accuracy))+geom_boxplot(fill="grey")+theme_minimal()+the
        me(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Accuracy")
plot3=ggplot(Data_Eval,aes(x=Methods,y=Sensitivity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("")+ylab("Specificity")
plot4=ggplot(Data_Eval,aes(x=Methods,y=Specificity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Sensitivity")
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot2_5A.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot1, plot2 , ncol = 1, nrow = 2)
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot3_5A.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot3, plot4 , ncol = 1, nrow = 2)
dev.off()
write.csv(Eval_mean,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_mean_Sim5A.csv",row.names = FALSE)
write.csv(Eval_sd,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_sd_Sim5A.csv",row.names = FALSE)
write.csv(Data_Eval,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_Sim5A.csv",row.names = FALSE)
```

Lampiran 7 : Sintaks R Studi Simulasi Skenario 6

```
library(caret)
library(e1071)
library(MLmetrics)
library(SwarmSVM)
library(fmsb)
library(ggpubr)
library(MASS)
n=500
iter=100
Eval_glm=matrix(nrow=iter,ncol=4)
Eval_svm=matrix(nrow=iter,ncol=4)
Evalkern_csvm2=matrix(nrow=iter,ncol=4)
Evalkmean_csvm2=matrix(nrow=iter,ncol=4)
Evalkern_csvm3=matrix(nrow=iter,ncol=4)
Evalkmean_csvm3=matrix(nrow=iter,ncol=4)
Evalkern_csvm4=matrix(nrow=iter,ncol=4)
Evalkmean_csvm4=matrix(nrow=iter,ncol=4)
for (i in 1:iter) {
 x1<- rnorm(n,20,5)
 x2<- rnorm(n,30,5)
 x<-data.frame(x1,x2)
 z <- 6*x$x1-4*x$x2
 pr <- exp(z) / ( 1 + exp(z))   ## inverse-logit
 y <- rbinom(n, 1, pr)
 df <- data.frame(y=y, x)
 #sum(pr)
 #sum(df$y)
 model_glm <- glm(y~-1+x1+x2,data=df,family = binomial())
 pred_glm <- ifelse(predict(model_glm,df,type="response")>0.5,1,0)
 Eval_glm[i,]<-
c(AUC(pred_glm,df$y),Accuracy(pred_glm,df$y),Sensitivity(pred_glm,df$y,positive="1"),Specificity(pre
d_glm,df$y,positive="1"))
 ModelSVM<-svm(y~-1+x1+x2,data=df,type="C-classification",gamma=0.7,cost=8)
 pred_svm<-predict(ModelSVM, df)
 Eval_svm[i,]<-
c(AUC(pred_svm,df$y),Accuracy(pred_svm,df$y),Sensitivity(pred_svm,df$y,positive="1"),Specificity(pr
ed_svm,df$y,positive="1"))
 #kernel_Kmean_SVM_2
 Modelkern_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm2 <- unlist(predict(Modelkern_csvm2, df[,-1]))
 Evalkern_csvm2[i,] <-
c(AUC(predkern_csvm2,df$y),Accuracy(predkern_csvm2,df$y),Sensitivity(predkern_csvm2,df$y,positive
="1"),Specificity(predkern_csvm2,df$y,positive="1"))
 #Kmean_SVM_2
 Modelkmean_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
 predkmean_csvm2 <- unlist(predict(Modelkmean_csvm2, df[,-1]))
 Evalkmean_csvm2[i,] <-
c(AUC(predkmean_csvm2,df$y),Accuracy(predkmean_csvm2,df$y),Sensitivity(predkmean_csvm2,df$y,p
ositive="1"),Specificity(predkmean_csvm2,df$y,positive="1"))
 #kernel_Kmean_SVM_3
 Modelkern_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm3 <- unlist(predict(Modelkern_csvm3, df[,-1]))
 Evalkern_csvm3[i,] <-
c(AUC(predkern_csvm3,df$y),Accuracy(predkern_csvm3,df$y),Sensitivity(predkern_csvm3,df$y,positive
="1"),Specificity(predkern_csvm3,df$y,positive="1"))
 #Kmean_SVM_3
 Modelkmean_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
 predkmean_csvm3 <- unlist(predict(Modelkmean_csvm3, df[,-1]))
```

```
  Evalkmean_csvm3[i,] <-
c(AUC(predkmean_csvm3,df$y),Accuracy(predkmean_csvm3,df$y),Sensitivity(predkmean_csvm3,df$y,p
ositive="1"),Specificity(predkmean_csvm3,df$y,positive="1"))
  #kernel_Kmean_SVM_4
  Modelkern_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 4, verbose =
0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm4 <- unlist(predict(Modelkern_csvm4, df[,-1]))
  Evalkern_csvm4[i,] <-
c(AUC(predkern_csvm4,df$y),Accuracy(predkern_csvm4,df$y),Sensitivity(predkern_csvm4,df$y,positive
="1"),Specificity(predkern_csvm4,df$y,positive="1"))
  #Kmean_SVM_4
  Modelkmean_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1, centers = 4, verbose = 0,
cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm4 <- unlist(predict(Modelkmean_csvm4, df[,-1]))
  Evalkmean_csvm4[i,] <-
c(AUC(predkmean_csvm4,df$y),Accuracy(predkmean_csvm4,df$y),Sensitivity(predkmean_csvm4,df$y,p
ositive="1"),Specificity(predkmean_csvm4,df$y,positive="1"))
}
Eval_mean=rbind(colMeans(Eval_glm),colMeans(Eval_svm),colMeans(Evalkern_csvm2),colMeans(Eval
kmean_csvm2),colMeans(Evalkern_csvm3),colMeans(Evalkmean_csvm3),colMeans(Evalkern_csvm4),co
lMeans(Evalkmean_csvm4))
Eval_sd=rbind(apply(Eval_glm, 2, sd),apply(Eval_svm, 2, sd),apply(Evalkern_csvm2, 2,
sd),apply(Evalkmean_csvm2, 2, sd),apply(Evalkern_csvm3, 2, sd),apply(Evalkmean_csvm3, 2,
sd),apply(Evalkern_csvm4, 2, sd),apply(Evalkmean_csvm4, 2, sd))
row.names(Eval_mean)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","ker
n_csvm4","kmean_svm4")
colnames(Eval_mean)=c("AUC","Accuracy","Sensitivity","Specificity")
row.names(Eval_sd)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","kern_c
svm4","kmean_svm4")
colnames(Eval_sd)=c("AUC","Accuracy","Sensitivity","Specificity")
Eval_mean
Eval_sd
Trans_Eval=t(Eval_mean)
data_new=as.data.frame(rbind(rep(1,8) , rep(0.95,8) , Trans_Eval))
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9),rgb(0.1,0.1,0.4,0.5) )
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Plot1_1B.jpeg",width=800,height = 600,quality=100)
radarchart( data_new , axistype=1 ,
        #custom polygon
        pcol=colors_border,
        #custom the grid
        cglcol="black", cglty=1, axislabcol="black", seg= 5, caxislabels=seq(0.95,1,0.01),
cglwd=0.4,plwd=3,plty=4,
        #custom labels
        vlcex=1.4)
legend(x=1, y=0.5, legend = rownames(data_new[-c(1,2),]), bty = "n", pch=20,col=colors_border,cex=1,
pt.cex=2.5)
dev.off()

Data_Eval=rbind(Eval_glm,Eval_svm,Evalkern_csvm2,Evalkmean_csvm2,Evalkern_csvm3,Evalkmean_
csvm3,Evalkern_csvm4,Evalkmean_csvm4)
Metode=c(rep("glm",iter),rep("svm",iter),rep("kern_csvm2",iter),rep("kmean_svm2",iter),rep("kern_csvm
3",iter),rep("kmean_svm3",iter),rep("kern_csvm4",iter),rep("kmean_csvm4",iter))
Data_Eval=data.frame(Data_Eval,Metode)
colnames(Data_Eval)=c("AUC","Accuracy","Sensitivity","Specificity","Methods")
plot1=ggplot(Data_Eval,aes(x=Methods,y=AUC))+geom_boxplot(fill="grey")+theme_minimal()+theme(
axis.text.x = element_text(size=16),panel.border = element_blank(),panel.grid=element_blank())+xlab("")
+ylab("AUC")
plot2=ggplot(Data_Eval,aes(x=Methods,y=Accuracy))+geom_boxplot(fill="grey")+theme_minimal()+the
me(axis.text.x = element_text(size=16),panel.border =
element_blank(),panel.grid=element_blank())+ylab("Accuracy")
```

```
plot3=ggplot(Data_Eval,aes(x=Methods,y=Sensitivity))+geom_boxplot(fill="grey")+theme_minimal()+th
eme(axis.text.x = element_text(size=16),panel.border =
element_blank(),panel.grid=element_blank())+xlab("")+ylab("Specificity")
plot4=ggplot(Data_Eval,aes(x=Methods,y=Specificity))+geom_boxplot(fill="grey")+theme_minimal()+th
eme(axis.text.x = element_text(size=16),panel.border =
element_blank(),panel.grid=element_blank())+ylab("Sensitivity")
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Plot2_1B.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot1, plot2 , ncol = 1, nrow = 2)
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Plot3_1B.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot3, plot4 , ncol = 1, nrow = 2)
dev.off()
write.csv(Eval_mean,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Hasil_mean_Sim1B.csv",row.names = FALSE)
write.csv(Eval_sd,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Hasil_sd_Sim1B.csv",row.names = FALSE)
write.csv(Data_Eval,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Hasil_Sim1B.csv",row.names = FALSE)
```

Lampiran 8 : Sintaks R Studi Simulasi Skenario 7

```
library(caret)
library(e1071)
library(MLmetrics)
library(SwarmSVM)
library(fmsb)
library(ggpubr)
library(MASS)

n=500
iter=100
Eval_glm=matrix(nrow=iter,ncol=4)
Eval_svm=matrix(nrow=iter,ncol=4)
Evalkern_csvm2=matrix(nrow=iter,ncol=4)
Evalkmean_csvm2=matrix(nrow=iter,ncol=4)
Evalkern_csvm3=matrix(nrow=iter,ncol=4)
Evalkmean_csvm3=matrix(nrow=iter,ncol=4)
Evalkern_csvm4=matrix(nrow=iter,ncol=4)
Evalkmean_csvm4=matrix(nrow=iter,ncol=4)
for (i in 1:iter) {
  x1_1<- rnorm(n/2,20,5)
  x1_2<- rnorm(n/2,50,5)
  x1<-c(x1_1,x1_2)
  x2_1<- rnorm(n/2,10,5)
  x2_2<- rnorm(n/2,30,5)
  x2<-c(x2_1,x2_2)
  x<-data.frame(x1,x2)
  z <- 3*x$x1-5*x$x2
  pr <- exp(z) / ( 1 + exp(z))   ## inverse-logit
  y <- rbinom(n, 1, pr)
  df <- data.frame(y=y, x)
  #sum(pr)
  #sum(df$y)
  model_glm <- glm(y~-1+x1+x2,data=df,family = binomial())
  pred_glm <- ifelse(predict(model_glm,df,type="response")>0.5,1,0)
  Eval_glm[i,]<-
c(AUC(pred_glm,df$y),Accuracy(pred_glm,df$y),Sensitivity(pred_glm,df$y,positive="1"),Specificity(pre
d_glm,df$y,positive="1"))
  ModelSVM<-svm(y~-1+x1+x2,data=df,type="C-classification",gamma=0.7,cost=8)
  pred_svm<-predict(ModelSVM, df)
  Eval_svm[i,]<-
c(AUC(pred_svm,df$y),Accuracy(pred_svm,df$y),Sensitivity(pred_svm,df$y,positive="1"),Specificity(pr
ed_svm,df$y,positive="1"))
  #kernel_Kmean_SVM_2
  Modelkern_csvm2 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 2, verbose =
0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm2 <- unlist(predict(Modelkern_csvm2, df[,-1]))
  Evalkern_csvm2[i,] <-
c(AUC(predkern_csvm2,df$y),Accuracy(predkern_csvm2,df$y),Sensitivity(predkern_csvm2,df$y,positive
="1"),Specificity(predkern_csvm2,df$y,positive="1"))
  #Kmean_SVM_2
  Modelkmean_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm2 <- unlist(predict(Modelkmean_csvm2, df[,-1]))
  Evalkmean_csvm2[i,] <-
c(AUC(predkmean_csvm2,df$y),Accuracy(predkmean_csvm2,df$y),Sensitivity(predkmean_csvm2,df$y,p
ositive="1"),Specificity(predkmean_csvm2,df$y,positive="1"))
  #kernel_Kmean_SVM_3
  Modelkern_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm3 <- unlist(predict(Modelkern_csvm3, df[,-1]))
```

```
  Evalkern_csvm3[i,] <-
c(AUC(predkern_csvm3,df$y),Accuracy(predkern_csvm3,df$y),Sensitivity(predkern_csvm3,df$y,positive
="1"),Specificity(predkern_csvm3,df$y,positive="1"))
  #Kmean_SVM_3
  Modelkmean_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm3 <- unlist(predict(Modelkmean_csvm3, df[,-1]))
  Evalkmean_csvm3[i,] <-
c(AUC(predkmean_csvm3,df$y),Accuracy(predkmean_csvm3,df$y),Sensitivity(predkmean_csvm3,df$y,p
ositive="1"),Specificity(predkmean_csvm3,df$y,positive="1"))
  #kernel_Kmean_SVM_4
  Modelkern_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 4, verbose =
0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm4 <- unlist(predict(Modelkern_csvm4, df[,-1]))
  Evalkern_csvm4[i,] <-
c(AUC(predkern_csvm4,df$y),Accuracy(predkern_csvm4,df$y),Sensitivity(predkern_csvm4,df$y,positive
="1"),Specificity(predkern_csvm4,df$y,positive="1"))
  #Kmean_SVM_4
  Modelkmean_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1, centers = 4, verbose = 0,
cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm4 <- unlist(predict(Modelkmean_csvm4, df[,-1]))
  Evalkmean_csvm4[i,] <-
c(AUC(predkmean_csvm4,df$y),Accuracy(predkmean_csvm4,df$y),Sensitivity(predkmean_csvm4,df$y,p
ositive="1"),Specificity(predkmean_csvm4,df$y,positive="1"))
}
Eval_mean=rbind(colMeans(Eval_glm),colMeans(Eval_svm),colMeans(Evalkern_csvm2),colMeans(Eval
kmean_csvm2),colMeans(Evalkern_csvm3),colMeans(Evalkmean_csvm3),colMeans(Evalkern_csvm4),co
lMeans(Evalkmean_csvm4))
Eval_sd=rbind(apply(Eval_glm, 2, sd),apply(Eval_svm, 2, sd),apply(Evalkern_csvm2, 2,
sd),apply(Evalkmean_csvm2, 2, sd),apply(Evalkern_csvm3, 2, sd),apply(Evalkmean_csvm3, 2,
sd),apply(Evalkern_csvm4, 2, sd),apply(Evalkmean_csvm4, 2, sd))
row.names(Eval_mean)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","ker
n_csvm4","kmean_svm4")
colnames(Eval_mean)=c("AUC","Accuracy","Sensitivity","Specificity")
row.names(Eval_sd)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","kern_c
svm4","kmean_svm4")
colnames(Eval_sd)=c("AUC","Accuracy","Sensitivity","Specificity")
Eval_mean
Eval_sd
Trans_Eval=t(Eval_mean)
data_new=as.data.frame(rbind(rep(1,8) , rep(0.95,8) , Trans_Eval))
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9),rgb(0.1,0.1,0.4,0.5) )
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Plot1_2B.jpeg",width=800,height = 600,quality=100)
radarchart( data_new , axistype=1 ,
        #custom polygon
        pcol=colors_border,
        #custom the grid
        cglcol="black", cglty=1, axislabcol="black", seg= 5, caxislabels=seq(0.95,1,0.01),
cglwd=0.4,plwd=3,plty=4,
        #custom labels
        vlcex=1.4)
legend(x=1, y=0.5, legend = rownames(data_new[-c(1,2),]), bty = "n", pch=20,col=colors_border,cex=1,
pt.cex=2.5)
dev.off()

Data_Eval=rbind(Eval_glm,Eval_svm,Evalkern_csvm2,Evalkmean_csvm2,Evalkern_csvm3,Evalkmean_
csvm3,Evalkern_csvm4,Evalkmean_csvm4)
Metode=c(rep("glm",iter),rep("svm",iter),rep("kern_csvm2",iter),rep("kmean_svm2",iter),rep("kern_csvm
3",iter),rep("kmean_svm3",iter),rep("kern_csvm4",iter),rep("kmean_csvm4",iter))
Data_Eval=data.frame(Data_Eval,Metode)
colnames(Data_Eval)=c("AUC","Accuracy","Sensitivity","Specificity","Methods")
```

```
plot1=ggplot(Data_Eval,aes(x=Methods,y=AUC))+geom_boxplot(fill="grey")+theme_minimal()+theme(
axis.text.x = element_text(size=16),panel.border = element_blank(),panel.grid=element_blank())+xlab("")
+ylab("AUC")
plot2=ggplot(Data_Eval,aes(x=Methods,y=Accuracy))+geom_boxplot(fill="grey")+theme_minimal()+the
me(axis.text.x = element_text(size=16),panel.border =
element_blank(),panel.grid=element_blank())+ylab("Accuracy")
plot3=ggplot(Data_Eval,aes(x=Methods,y=Sensitivity))+geom_boxplot(fill="grey")+theme_minimal()+th
eme(axis.text.x = element_text(size=16),panel.border =
element_blank(),panel.grid=element_blank())+xlab("")+ylab("Specificity")
plot4=ggplot(Data_Eval,aes(x=Methods,y=Specificity))+geom_boxplot(fill="grey")+theme_minimal()+th
eme(axis.text.x = element_text(size=16),panel.border =
element_blank(),panel.grid=element_blank())+ylab("Sensitivity")
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Plot2_2B.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot1, plot2 , ncol = 1, nrow = 2)
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Plot3_2B.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot3, plot4 , ncol = 1, nrow = 2)
dev.off()
write.csv(Eval_mean,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Hasil_mean_Sim2B.csv",row.names = FALSE)
write.csv(Eval_sd,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Hasil_sd_Sim2B.csv",row.names = FALSE)
write.csv(Data_Eval,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Simulasi/Hasil_Sim2B.csv",row.names = FALSE)
```

Lampiran 9 : Sintaks R Studi Simulasi Skenario 8

```
library(caret)
library(e1071)
library(MLmetrics)
library(SwarmSVM)
library(MASS)
library(ggpubr)
library(fmsb)

n=500
iter=100
Eval_glm=matrix(nrow=iter,ncol=4)
Eval_svm=matrix(nrow=iter,ncol=4)
Evalkern_csvm2=matrix(nrow=iter,ncol=4)
Evalkmean_csvm2=matrix(nrow=iter,ncol=4)
Evalkern_csvm3=matrix(nrow=iter,ncol=4)
Evalkmean_csvm3=matrix(nrow=iter,ncol=4)
Evalkern_csvm4=matrix(nrow=iter,ncol=4)
Evalkmean_csvm4=matrix(nrow=iter,ncol=4)
mu_1=c(20,40)
mu_2=c(50,70)
sigma_1=matrix(c(25,16,16,25),2,2)
sigma_2=matrix(c(25,16,16,25),2,2)
for (i in 1:iter) {
 x1_1<- rnorm(n/2,20,5)
 x1_2<- rnorm(n/2,50,5)
 x1<-c(x1_1,x1_2)
 x2_1<- rnorm(n/2,10,5)
 x2_2<- rnorm(n/2,30,5)
 x2<-c(x2_1,x2_2)
 x3_x4_1 <- mvrnorm(n/2, mu_1, sigma_1, empirical = TRUE)
 colnames(x3_x4_1)=c("x3","x4")
 x3_x4_2 <- mvrnorm(n/2, mu_2, sigma_2, empirical = TRUE)
 colnames(x3_x4_2)=c("x3","x4")
 x3_x4<-rbind(x3_x4_1,x3_x4_2)
 x3<-x3_x4[,1]
 x4<-x3_x4[,2]
 x<-data.frame(x1,x2,x3,x4)
 z <- 4*x$x1-5*x$x2-6*x$x3+3*x$x4
 pr <- exp(z) / ( 1 + exp(z))   ## inverse-logit
 y <- rbinom(n, 1, pr)
 df <- data.frame(y=y, x)
 #sum(pr)
 #sum(df$y)
 model_glm <- glm(y~-1+x1+x2+x3+x4,data=df,family = binomial())
 pred_glm <- ifelse(predict(model_glm,df,type="response")>0.5,1,0)
 Eval_glm[i,]<-
       c(AUC(pred_glm,df$y),Accuracy(pred_glm,df$y),Sensitivity(pred_glm,df$y,positive="1"),Specifi
       city(pred_glm,df$y,positive="1"))
 ModelSVM<-svm(y~-1+x1+x2+x3+x4,data=df,type="C-classification",gamma=0.7,cost=8)
 pred_svm<-predict(ModelSVM, df)
 Eval_svm[i,]<-
       c(AUC(pred_svm,df$y),Accuracy(pred_svm,df$y),Sensitivity(pred_svm,df$y,positive="1"),Specif
       icity(pred_svm,df$y,positive="1"))
 #kernel_Kmean_SVM_2
 Modelkern_csvm2 <- clusterSVM(x = df[,-1], y = df[,1], lambda = 1,centers = 2,cost=8, verbose =
       0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm2 <- unlist(predict(Modelkern_csvm2, df[,-1]))
 Evalkern_csvm2[i,] <-
       c(AUC(predkern_csvm2,df$y),Accuracy(predkern_csvm2,df$y),Sensitivity(predkern_csvm2,df$y,
       positive="1"),Specificity(predkern_csvm2,df$y,positive="1"))
 #Kmean_SVM_2
```

```
  Modelkmean_csvm2 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 2, verbose =
        0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm2 <- unlist(predict(Modelkmean_csvm2, df[,-1]))
  Evalkmean_csvm2[i,] <-
        c(AUC(predkmean_csvm2,df$y),Accuracy(predkmean_csvm2,df$y),Sensitivity(predkmean_csvm
        2,df$y,positive="1"),Specificity(predkmean_csvm2,df$y,positive="1"))
  #kernel_Kmean_SVM_3
  Modelkern_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm3 <- unlist(predict(Modelkern_csvm3, df[,-1]))
  Evalkern_csvm3[i,] <-
        c(AUC(predkern_csvm3,df$y),Accuracy(predkern_csvm3,df$y),Sensitivity(predkern_csvm3,df$y,
        positive="1"),Specificity(predkern_csvm3,df$y,positive="1"))
  #Kmean_SVM_3
  Modelkmean_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
        0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm3 <- unlist(predict(Modelkmean_csvm3, df[,-1]))
  Evalkmean_csvm3[i,] <-
        c(AUC(predkmean_csvm3,df$y),Accuracy(predkmean_csvm3,df$y),Sensitivity(predkmean_csvm
        3,df$y,positive="1"),Specificity(predkmean_csvm3,df$y,positive="1"))
  #kernel_Kmean_SVM_4
  Modelkern_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 4, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm4 <- unlist(predict(Modelkern_csvm4, df[,-1]))
  Evalkern_csvm4[i,] <-
        c(AUC(predkern_csvm4,df$y),Accuracy(predkern_csvm4,df$y),Sensitivity(predkern_csvm4,df$y,
        positive="1"),Specificity(predkern_csvm4,df$y,positive="1"))
  #Kmean_SVM_4
  Modelkmean_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1, centers = 4, verbose = 0,
        cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm4 <- unlist(predict(Modelkmean_csvm4, df[,-1]))
  Evalkmean_csvm4[i,] <-
        c(AUC(predkmean_csvm4,df$y),Accuracy(predkmean_csvm4,df$y),Sensitivity(predkmean_csvm
        4,df$y,positive="1"),Specificity(predkmean_csvm4,df$y,positive="1"))
}
Eval_mean=rbind(colMeans(Eval_glm),colMeans(Eval_svm),colMeans(Evalkern_csvm2),colMeans(Eval
        kmean_csvm2),colMeans(Evalkern_csvm3),colMeans(Evalkmean_csvm3),colMeans(Evalkern_cs
        vm4),colMeans(Evalkmean_csvm4))
Eval_sd=rbind(apply(Eval_glm, 2, sd),apply(Eval_svm, 2, sd),apply(Evalkern_csvm2, 2,
        sd),apply(Evalkmean_csvm2, 2, sd),apply(Evalkern_csvm3, 2, sd),apply(Evalkmean_csvm3, 2,
        sd),apply(Evalkern_csvm4, 2, sd),apply(Evalkmean_csvm4, 2, sd))
row.names(Eval_mean)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","ker
        n_csvm4","kmean_svm4")
colnames(Eval_mean)=c("AUC","Accuracy","Sensitivity","Specificity")
row.names(Eval_sd)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","kern_c
        svm4","kmean_svm4")
colnames(Eval_sd)=c("AUC","Accuracy","Sensitivity","Specificity")
Eval_mean
Eval_sd
Trans_Eval=t(Eval_mean)
data_new=as.data.frame(rbind(rep(1,8) , rep(0.95,8) , Trans_Eval))
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9),rgb(0.1,0.1,0.4,0.5) )
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot1_3B.jpeg",width=800,height = 600,quality=100)
radarchart( data_new , axistype=1 ,
        #custom polygon
        pcol=colors_border,
        #custom the grid
        cglcol="black", cglty=1, axislabcol="black", seg= 5, caxislabels=seq(0.95,1,0.01),
        cglwd=0.4,plwd=3,plty=4,
        #custom labels
        vlcex=1.4)
```

```
legend(x=1, y=0.5, legend = rownames(data_new[-c(1,2),]), bty = "n", pch=20,col=colors_border,cex=1,
        pt.cex=2.5)
dev.off()

Data_Eval=rbind(Eval_glm,Eval_svm,Evalkern_csvm2,Evalkmean_csvm2,Evalkern_csvm3,Evalkmean_
        csvm3,Evalkern_csvm4,Evalkmean_csvm4)
Metode=c(rep("glm",iter),rep("svm",iter),rep("kern_csvm2",iter),rep("kmean_svm2",iter),rep("kern_csvm
        3",iter),rep("kmean_svm3",iter),rep("kern_csvm4",iter),rep("kmean_csvm4",iter))
Data_Eval=data.frame(Data_Eval,Metode)
colnames(Data_Eval)=c("AUC","Accuracy","Sensitivity","Specificity","Methods")
plot1=ggplot(Data_Eval,aes(x=Methods,y=AUC))+geom_boxplot(fill="grey")+theme_minimal()+theme(
        axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("") +ylab("AUC")
plot2=ggplot(Data_Eval,aes(x=Methods,y=Accuracy))+geom_boxplot(fill="grey")+theme_minimal()+the
        me(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Accuracy")
plot3=ggplot(Data_Eval,aes(x=Methods,y=Sensitivity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("")+ylab("Specificity")
plot4=ggplot(Data_Eval,aes(x=Methods,y=Specificity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Sensitivity")
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot2_3B.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot1, plot2 , ncol = 1, nrow = 2)
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot3_3B.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot3, plot4 , ncol = 1, nrow = 2)
dev.off()
write.csv(Eval_mean,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_mean_Sim3B.csv",row.names = FALSE)
write.csv(Eval_sd,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_sd_Sim3B.csv",row.names = FALSE)
write.csv(Data_Eval,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_Sim3B.csv",row.names = FALSE)
```

Lampiran 10 : Sintaks R Studi Simulasi Skenario 9

```
library(caret)
library(e1071)
library(MLmetrics)
library(SwarmSVM)
library(MASS)
library(fmsb)

n=500
iter=100
Eval_glm=matrix(nrow=iter,ncol=4)
Eval_svm=matrix(nrow=iter,ncol=4)
Evalkern_csvm2=matrix(nrow=iter,ncol=4)
Evalkmean_csvm2=matrix(nrow=iter,ncol=4)
Evalkern_csvm3=matrix(nrow=iter,ncol=4)
Evalkmean_csvm3=matrix(nrow=iter,ncol=4)
Evalkern_csvm4=matrix(nrow=iter,ncol=4)
Evalkmean_csvm4=matrix(nrow=iter,ncol=4)
for (i in 1:iter) {
 x1_1<- rnorm(n/2,20,5)
 x1_2<- rnorm(n/2,50,5)
 x1<-c(x1_1,x1_2)
 x2_1<- rnorm(n/2,10,5)
 x2_2<- rnorm(n/2,30,5)
 x2<-c(x2_1,x2_2)
 x3<-5*x1-3*x2
 x<-data.frame(x1,x2,x3)
 z <- 4*x$x1+4.5*x$x2-2*x$x3
 pr <- exp(z) / ( 1 + exp(z))  ## inverse-logit
 y <- rbinom(n, 1, pr)
 df <- data.frame(y=y, x)
 model_glm <- glm(y~-1+x1+x2+x3,data=df,family = binomial())
 pred_glm <- ifelse(predict(model_glm,df,type="response")>0.5,1,0)
 Eval_glm[i,]<-
      c(AUC(pred_glm,df$y),Accuracy(pred_glm,df$y),Sensitivity(pred_glm,df$y,positive="1"),Specifi
      city(pred_glm,df$y,positive="1"))
 ModelSVM<-svm(y~-1+x1+x2+x3,data=df,type="C-classification",gamma=0.7,cost=8)
 pred_svm<-predict(ModelSVM, df)
 Eval_svm[i,]<-
      c(AUC(pred_svm,df$y),Accuracy(pred_svm,df$y),Sensitivity(pred_svm,df$y,positive="1"),Specif
      icity(pred_svm,df$y,positive="1"))
 #kernel_Kmean_SVM_2
 Modelkern_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
      0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm2 <- unlist(predict(Modelkern_csvm2, df[,-1]))
 Evalkern_csvm2[i,] <-
      c(AUC(predkern_csvm2,df$y),Accuracy(predkern_csvm2,df$y),Sensitivity(predkern_csvm2,df$y,
      positive="1"),Specificity(predkern_csvm2,df$y,positive="1"))
 #Kmean_SVM_2
 Modelkmean_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
      0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
 predkmean_csvm2 <- unlist(predict(Modelkmean_csvm2, df[,-1]))
 Evalkmean_csvm2[i,] <-
      c(AUC(predkmean_csvm2,df$y),Accuracy(predkmean_csvm2,df$y),Sensitivity(predkmean_csvm
      2,df$y,positive="1"),Specificity(predkmean_csvm2,df$y,positive="1"))
 #kernel_Kmean_SVM_3
 Modelkern_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
      0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm3 <- unlist(predict(Modelkern_csvm3, df[,-1]))
 Evalkern_csvm3[i,] <-
      c(AUC(predkern_csvm3,df$y),Accuracy(predkern_csvm3,df$y),Sensitivity(predkern_csvm3,df$y,
      positive="1"),Specificity(predkern_csvm3,df$y,positive="1"))
 #Kmean_SVM_3
```

```
  Modelkmean_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
        0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm3 <- unlist(predict(Modelkmean_csvm3, df[,-1]))
  Evalkmean_csvm3[i,] <-
        c(AUC(predkmean_csvm3,df$y),Accuracy(predkmean_csvm3,df$y),Sensitivity(predkmean_csvm
        3,df$y,positive="1"),Specificity(predkmean_csvm3,df$y,positive="1"))
  #kernel_Kmean_SVM_4
  Modelkern_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 4, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm4 <- unlist(predict(Modelkern_csvm4, df[,-1]))
  Evalkern_csvm4[i,] <-
        c(AUC(predkern_csvm4,df$y),Accuracy(predkern_csvm4,df$y),Sensitivity(predkern_csvm4,df$y,
        positive="1"),Specificity(predkern_csvm4,df$y,positive="1"))
  #Kmean_SVM_4
  Modelkmean_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1, centers = 4, verbose = 0,
        cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm4 <- unlist(predict(Modelkmean_csvm4, df[,-1]))
  Evalkmean_csvm4[i,] <-
        c(AUC(predkmean_csvm4,df$y),Accuracy(predkmean_csvm4,df$y),Sensitivity(predkmean_csvm
        4,df$y,positive="1"),Specificity(predkmean_csvm4,df$y,positive="1"))
}
Eval_mean=rbind(colMeans(Eval_glm),colMeans(Eval_svm),colMeans(Evalkern_csvm2),colMeans(Eval
        kmean_csvm2),colMeans(Evalkern_csvm3),colMeans(Evalkmean_csvm3),colMeans(Evalkern_cs
        vm4),colMeans(Evalkmean_csvm4))
Eval_sd=rbind(apply(Eval_glm, 2, sd),apply(Eval_svm, 2, sd),apply(Evalkern_csvm2, 2,
        sd),apply(Evalkmean_csvm2, 2, sd),apply(Evalkern_csvm3, 2, sd),apply(Evalkmean_csvm3, 2,
        sd),apply(Evalkern_csvm4, 2, sd),apply(Evalkmean_csvm4, 2, sd))
row.names(Eval_mean)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","ker
        n_csvm4","kmean_svm4")
colnames(Eval_mean)=c("AUC","Accuracy","Sensitivity","Specificity")
row.names(Eval_sd)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","kern_c
        svm4","kmean_svm4")
colnames(Eval_sd)=c("AUC","Accuracy","Sensitivity","Specificity")
Eval_mean
Eval_sd
Trans_Eval=t(Eval_mean)
data_new=as.data.frame(rbind(rep(1,8) , rep(0.95,8) , Trans_Eval))
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9),rgb(0.1,0.1,0.4,0.5) )
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot1_4B.jpeg",width=800,height = 600,quality=100)
radarchart( data_new , axistype=1 ,
        #custom polygon
        pcol=colors_border,
        #custom the grid
        cglcol="black", cglty=1, axislabcol="black", seg= 5, caxislabels=seq(0.95,1,0.01),
        cglwd=0.4,plwd=3,plty=4,
        #custom labels
        vlcex=1.4)
legend(x=1, y=0.5, legend = rownames(data_new[-c(1,2),]), bty = "n", pch=20,col=colors_border,cex=1,
        pt.cex=2.5)
dev.off()

Data_Eval=rbind(Eval_glm,Eval_svm,Evalkern_csvm2,Evalkmean_csvm2,Evalkern_csvm3,Evalkmean_
        csvm3,Evalkern_csvm4,Evalkmean_csvm4)
Metode=c(rep("glm",iter),rep("svm",iter),rep("kern_csvm2",iter),rep("kmean_svm2",iter),rep("kern_csvm
        3",iter),rep("kmean_svm3",iter),rep("kern_csvm4",iter),rep("kmean_csvm4",iter))
Data_Eval=data.frame(Data_Eval,Metode)
colnames(Data_Eval)=c("AUC","Accuracy","Sensitivity","Specificity","Methods")
plot1=ggplot(Data_Eval,aes(x=Methods,y=AUC))+geom_boxplot(fill="grey")+theme_minimal()+theme(
        axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("") +ylab("AUC")
```

```
plot2=ggplot(Data_Eval,aes(x=Methods,y=Accuracy))+geom_boxplot(fill="grey")+theme_minimal()+the
        me(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Accuracy")
plot3=ggplot(Data_Eval,aes(x=Methods,y=Sensitivity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("")+ylab("Specificity")
plot4=ggplot(Data_Eval,aes(x=Methods,y=Specificity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Sensitivity")
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot2_4B.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot1, plot2 , ncol = 1, nrow = 2)
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot3_4B.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot3, plot4 , ncol = 1, nrow = 2)
dev.off()
write.csv(Eval_mean,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_mean_Sim4B.csv",row.names = FALSE)
write.csv(Eval_sd,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_sd_Sim4B.csv",row.names = FALSE)
write.csv(Data_Eval,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_Sim4B.csv",row.names = FALSE)
```

Lampiran 11 : Sintaks R Studi Simulasi Skenario 10

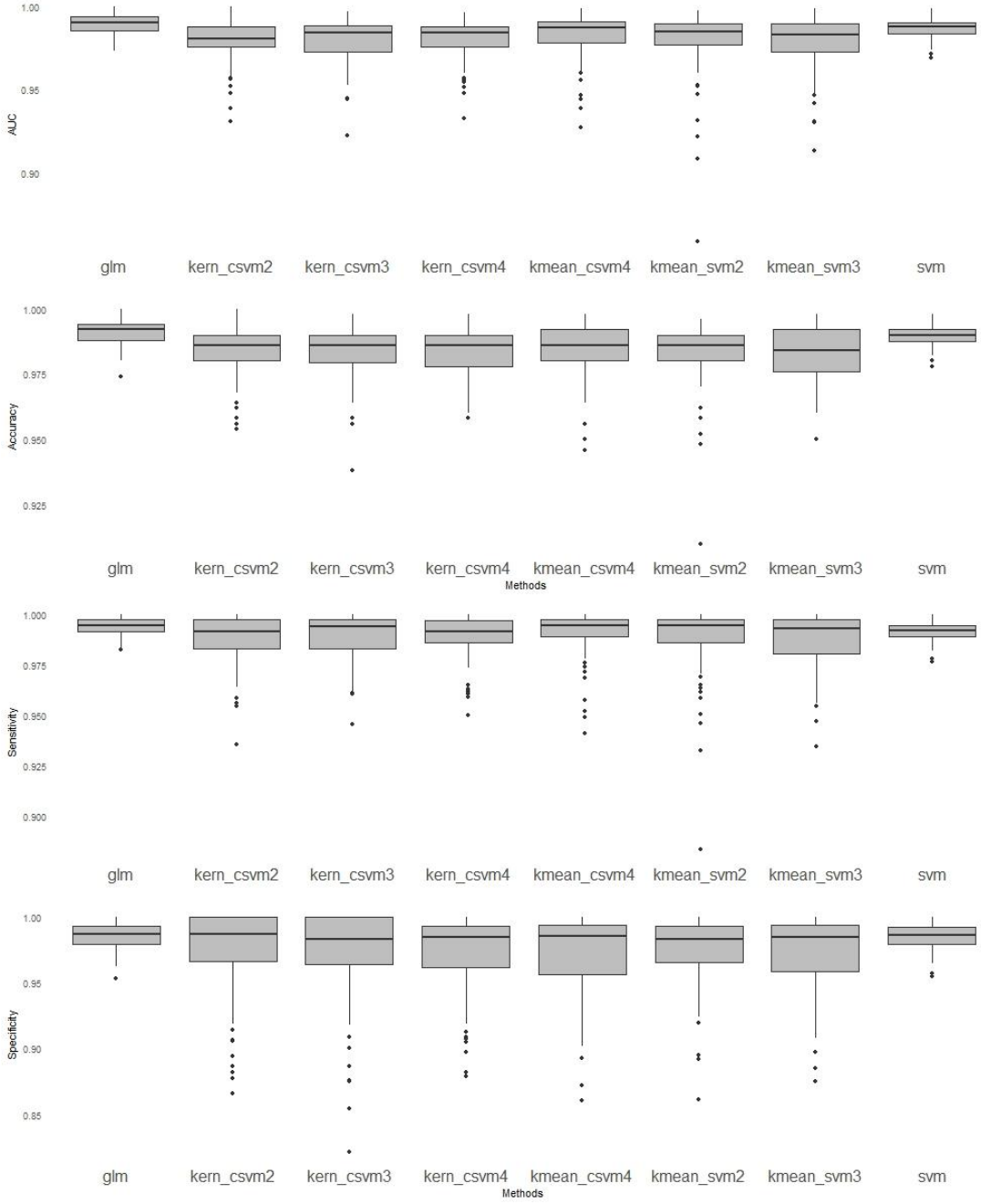
```
library(caret)
library(e1071)
library(MLmetrics)
library(SwarmSVM)

n=500
iter=100
Eval_glm=matrix(nrow=iter,ncol=4)
Eval_svm=matrix(nrow=iter,ncol=4)
Evalkern_csvm2=matrix(nrow=iter,ncol=4)
Evalkmean_csvm2=matrix(nrow=iter,ncol=4)
Evalkern_csvm3=matrix(nrow=iter,ncol=4)
Evalkmean_csvm3=matrix(nrow=iter,ncol=4)
Evalkern_csvm4=matrix(nrow=iter,ncol=4)
Evalkmean_csvm4=matrix(nrow=iter,ncol=4)
mu_1=c(20,40)
mu_2=c(10,10)
sigma_1=matrix(c(25,16,16,25),2,2)
sigma_2=matrix(c(25,16,16,25),2,2)
for (i in 1:iter) {
 x1_1<- rnorm(n/2,20,5)
 x1_2<- rnorm(n/2,50,5)
 x1<-c(x1_1,x1_2)
 x2_1<- rnorm(n/2,10,5)
 x2_2<- rnorm(n/2,30,5)
 x2<-c(x2_1,x2_2)
 x3<-5*x1-3*x2
 x4_x5_1 <- mvrnorm(n/2, mu_1, sigma_1, empirical = TRUE)
 colnames(x4_x5_1)=c("x4","x5")
 x4_x5_2 <- mvrnorm(n/2, mu_2, sigma_2, empirical = TRUE)
 colnames(x4_x5_2)=c("x4","x5")
 x4_x5<-rbind(x4_x5_1,x4_x5_2)
 x4<-x4_x5[,1]
 x5<-x4_x5[,2]
 x<-data.frame(x1,x2,x3,x4,x5)
 z <- 6*x$x1+6*x$x2-3*x$x3+5*x$x4-2*x$x5
 pr <- exp(z) / ( 1 + exp(z))   ## inverse-logit
 y <- rbinom(n, 1, pr)
 df <- data.frame(y=y, x)
 #sum(pr)
 #sum(df$y)
 model_glm <- glm(y~-1+x1+x2+x3+x4+x5,data=df,family = binomial())
 pred_glm <- ifelse(predict(model_glm,df,type="response")>0.5,1,0)
 Eval_glm[i,]<-
       c(AUC(pred_glm,df$y),Accuracy(pred_glm,df$y),Sensitivity(pred_glm,df$y,positive="1"),Specifi
       city(pred_glm,df$y,positive="1"))
 ModelSVM<-svm(y~-1+x1+x2+x3+x4+x5,data=df,type="C-classification",gamma=0.7,cost=8)
 pred_svm<-predict(ModelSVM, df)
 Eval_svm[i,]<-
       c(AUC(pred_svm,df$y),Accuracy(pred_svm,df$y),Sensitivity(pred_svm,df$y,positive="1"),Specif
       icity(pred_svm,df$y,positive="1"))
 #kernel_Kmean_SVM_2
 Modelkern_csvm2 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 2, verbose =
       0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
 predkern_csvm2 <- unlist(predict(Modelkern_csvm2, df[,-1]))
 Evalkern_csvm2[i,] <-
       c(AUC(predkern_csvm2,df$y),Accuracy(predkern_csvm2,df$y),Sensitivity(predkern_csvm2,df$y,
       positive="1"),Specificity(predkern_csvm2,df$y,positive="1"))
 #Kmean_SVM_2
 Modelkmean_csvm2 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 2, verbose =
       0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
 predkmean_csvm2 <- unlist(predict(Modelkmean_csvm2, df[,-1]))
```

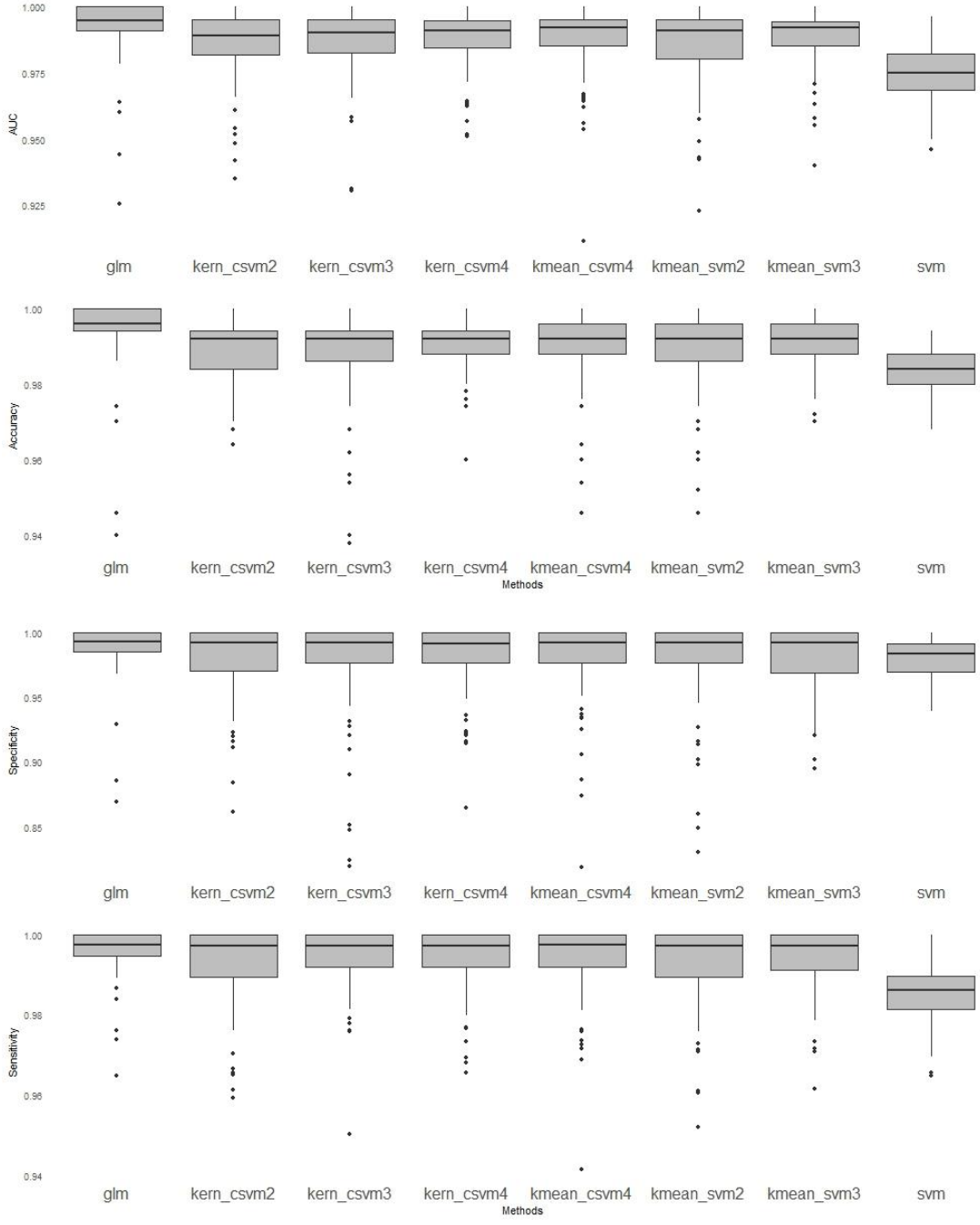
```
  Evalkmean_csvm2[i,] <-
        c(AUC(predkmean_csvm2,df$y),Accuracy(predkmean_csvm2,df$y),Sensitivity(predkmean_csvm
        2,df$y,positive="1"),Specificity(predkmean_csvm2,df$y,positive="1"))
  #kernel_Kmean_SVM_3
  Modelkern_csvm3 <- clusterSVM(x = df[,-1], y = df[,1],cost=8, lambda = 1,centers = 3, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm3 <- unlist(predict(Modelkern_csvm3, df[,-1]))
  Evalkern_csvm3[i,] <-
        c(AUC(predkern_csvm3,df$y),Accuracy(predkern_csvm3,df$y),Sensitivity(predkern_csvm3,df$y,
        positive="1"),Specificity(predkern_csvm3,df$y,positive="1"))
  #Kmean_SVM_3
  Modelkmean_csvm3 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 3, verbose =
        0,cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm3 <- unlist(predict(Modelkmean_csvm3, df[,-1]))
  Evalkmean_csvm3[i,] <-
        c(AUC(predkmean_csvm3,df$y),Accuracy(predkmean_csvm3,df$y),Sensitivity(predkmean_csvm
        3,df$y,positive="1"),Specificity(predkmean_csvm3,df$y,positive="1"))
  #kernel_Kmean_SVM_4
  Modelkern_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1,centers = 4, verbose =
        0,cluster.method="kernkmeans",valid.x = df[,-1],valid.y = df[,1])
  predkern_csvm4 <- unlist(predict(Modelkern_csvm4, df[,-1]))
  Evalkern_csvm4[i,] <-
        c(AUC(predkern_csvm4,df$y),Accuracy(predkern_csvm4,df$y),Sensitivity(predkern_csvm4,df$y,
        positive="1"),Specificity(predkern_csvm4,df$y,positive="1"))
  #Kmean_SVM_4
  Modelkmean_csvm4 <- clusterSVM(x = df[,-1], y = df[,1], cost=8,lambda = 1, centers = 4, verbose = 0,
        cluster.method="kmeans",valid.x = df[,-1],valid.y = df[,1])
  predkmean_csvm4 <- unlist(predict(Modelkmean_csvm4, df[,-1]))
  Evalkmean_csvm4[i,] <-
        c(AUC(predkmean_csvm4,df$y),Accuracy(predkmean_csvm4,df$y),Sensitivity(predkmean_csvm
        4,df$y,positive="1"),Specificity(predkmean_csvm4,df$y,positive="1"))
}
Eval_mean=rbind(colMeans(Eval_glm),colMeans(Eval_svm),colMeans(Evalkern_csvm2),colMeans(Eval
        kmean_csvm2),colMeans(Evalkern_csvm3),colMeans(Evalkmean_csvm3),colMeans(Evalkern_cs
        vm4),colMeans(Evalkmean_csvm4))
Eval_sd=rbind(apply(Eval_glm, 2, sd),apply(Eval_svm, 2, sd),apply(Evalkern_csvm2, 2,
        sd),apply(Evalkmean_csvm2, 2, sd),apply(Evalkern_csvm3, 2, sd),apply(Evalkmean_csvm3, 2,
        sd),apply(Evalkern_csvm4, 2, sd),apply(Evalkmean_csvm4, 2, sd))
row.names(Eval_mean)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","ker
        n_csvm4","kmean_svm4")
colnames(Eval_mean)=c("AUC","Accuracy","Sensitivity","Specificity")
row.names(Eval_sd)=c("glm","svm","kern_csvm2","kmean_svm2","kern_csvm3","kmean_svm3","kern_c
        svm4","kmean_svm4")
colnames(Eval_sd)=c("AUC","Accuracy","Sensitivity","Specificity")
Eval_mean
Eval_sd
Trans_Eval=t(Eval_mean)
data_new=as.data.frame(rbind(rep(1,8) , rep(0.95,8) , Trans_Eval))
colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rgb(0.7,0.5,0.1,0.9),rgb(0.1,0.1,0.4,0.5) )
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot1_5B.jpeg",width=800,height = 600,quality=100)
radarchart( data_new , axistype=1 ,
        #custom polygon
        pcol=colors_border,
        #custom the grid
        cglcol="black", cglty=1, axislabcol="black", seg= 5, caxislabels=seq(0.95,1,0.01),
        cglwd=0.4,plwd=3,plty=4,
        #custom labels
        vlcex=1.4)
legend(x=1, y=0.5, legend = rownames(data_new[-c(1,2),]), bty = "n", pch=20,col=colors_border,cex=1,
        pt.cex=2.5)
dev.off()
```

```
Data_Eval=rbind(Eval_glm,Eval_svm,Evalkern_csvm2,Evalkmean_csvm2,Evalkern_csvm3,Evalkmean_
        csvm3,Evalkern_csvm4,Evalkmean_csvm4)
Metode=c(rep("glm",iter),rep("svm",iter),rep("kern_csvm2",iter),rep("kmean_svm2",iter),rep("kern_csvm
        3",iter),rep("kmean_svm3",iter),rep("kern_csvm4",iter),rep("kmean_csvm4",iter))
Data_Eval=data.frame(Data_Eval,Metode)
colnames(Data_Eval)=c("AUC","Accuracy","Sensitivity","Specificity","Methods")
plot1=ggplot(Data_Eval,aes(x=Methods,y=AUC))+geom_boxplot(fill="grey")+theme_minimal()+theme(
        axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("") +ylab("AUC")
plot2=ggplot(Data_Eval,aes(x=Methods,y=Accuracy))+geom_boxplot(fill="grey")+theme_minimal()+the
        me(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Accuracy")
plot3=ggplot(Data_Eval,aes(x=Methods,y=Sensitivity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+xlab("")+ylab("Specificity")
plot4=ggplot(Data_Eval,aes(x=Methods,y=Specificity))+geom_boxplot(fill="grey")+theme_minimal()+th
        eme(axis.text.x = element_text(size=16),panel.border =
        element_blank(),panel.grid=element_blank())+ylab("Sensitivity")
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot2_5B.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot1, plot2 , ncol = 1, nrow = 2)
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Plot3_5B.jpeg",width=1000,height = 600,quality=100)
ggarrange(plot3, plot4 , ncol = 1, nrow = 2)
dev.off()
write.csv(Eval_mean,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_mean_Sim5B.csv",row.names = FALSE)
write.csv(Eval_sd,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_sd_Sim5B.csv",row.names = FALSE)
write.csv(Data_Eval,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Simulasi/Hasil_Sim5B.csv",row.names = FALSE)
```

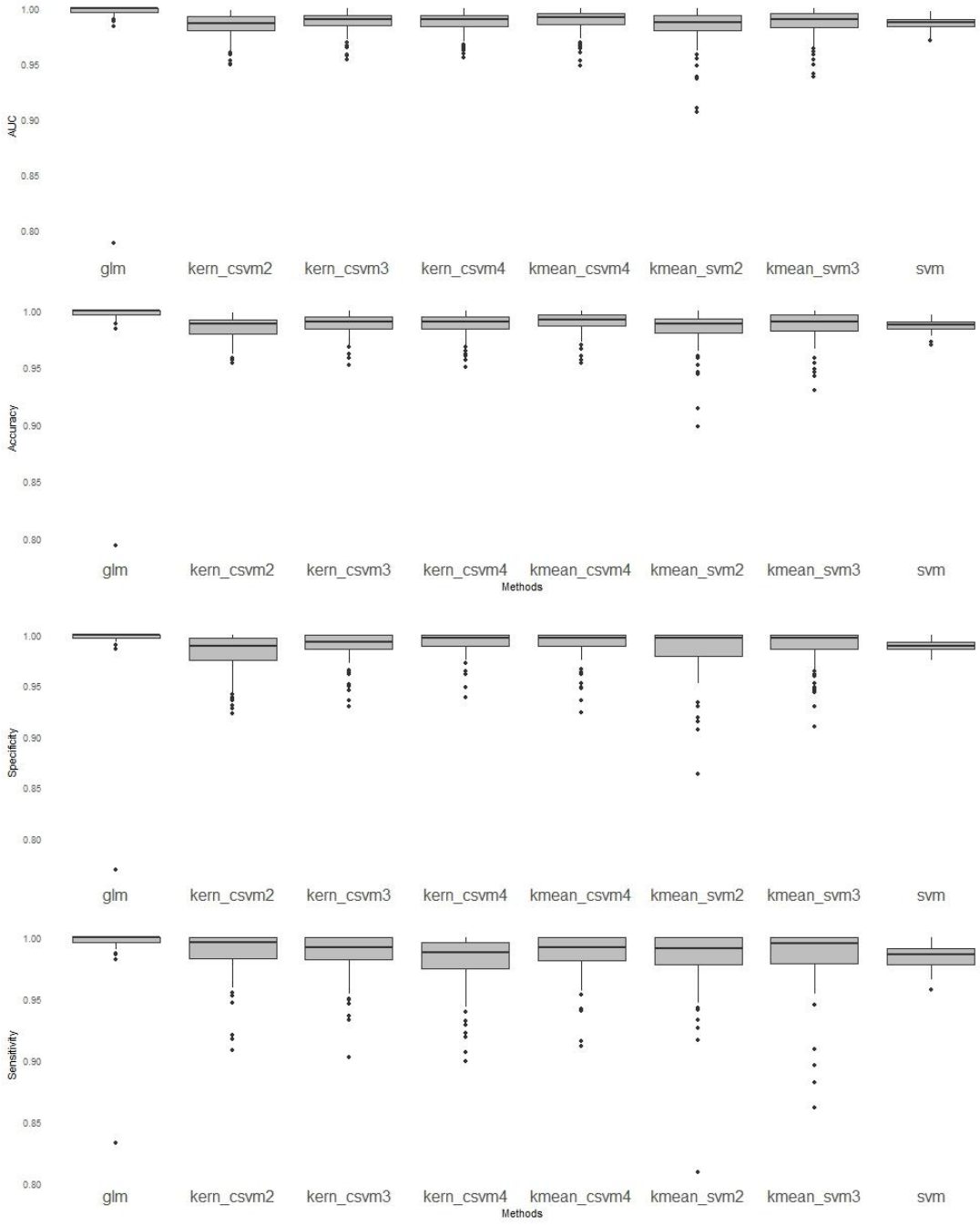
Lampiran 12 : Box Plot Evaluasi Model Studi Simulasi Skenario 1



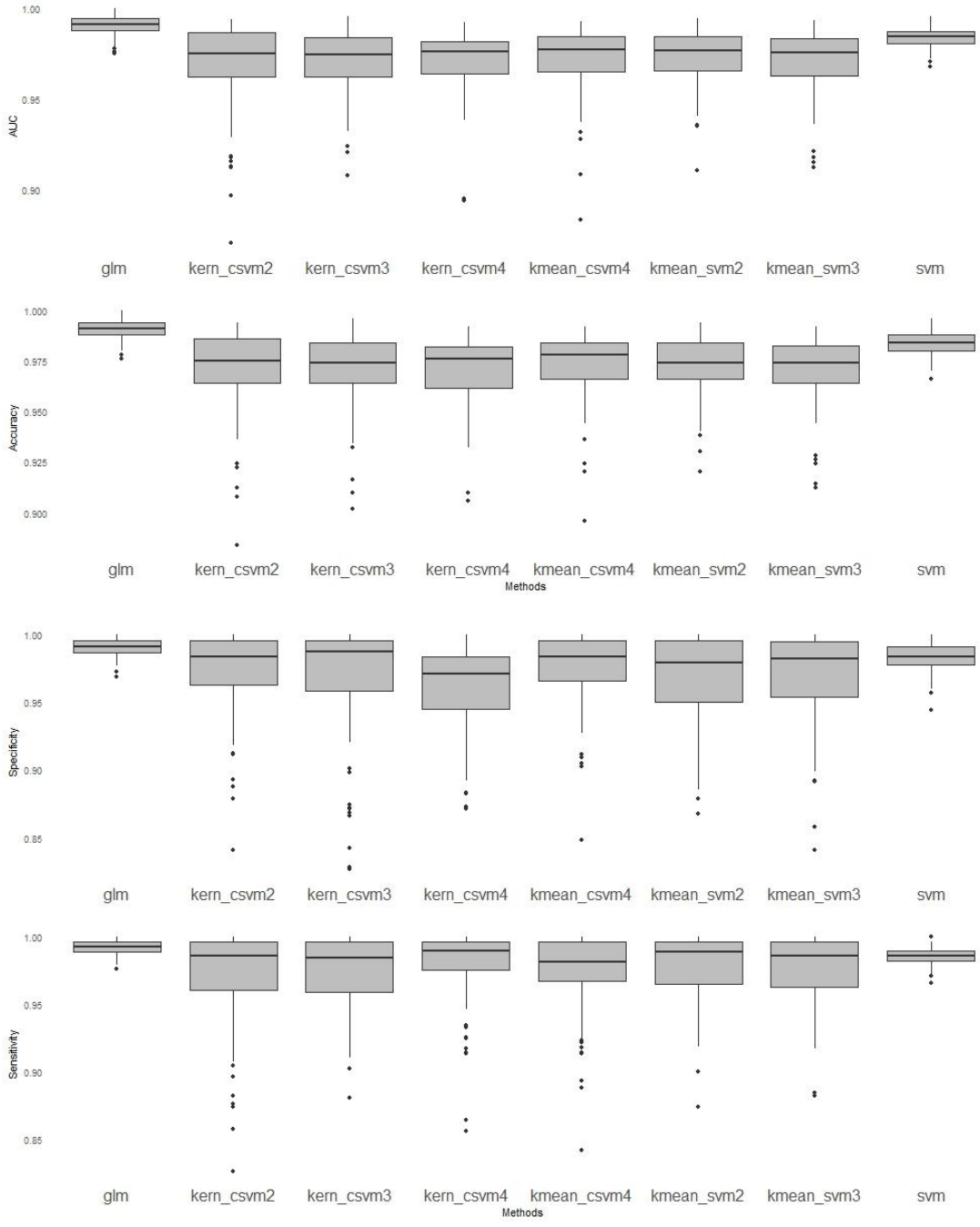
Lampiran 13 : Box Plot Evaluasi Model Studi Simulasi Skenario 2



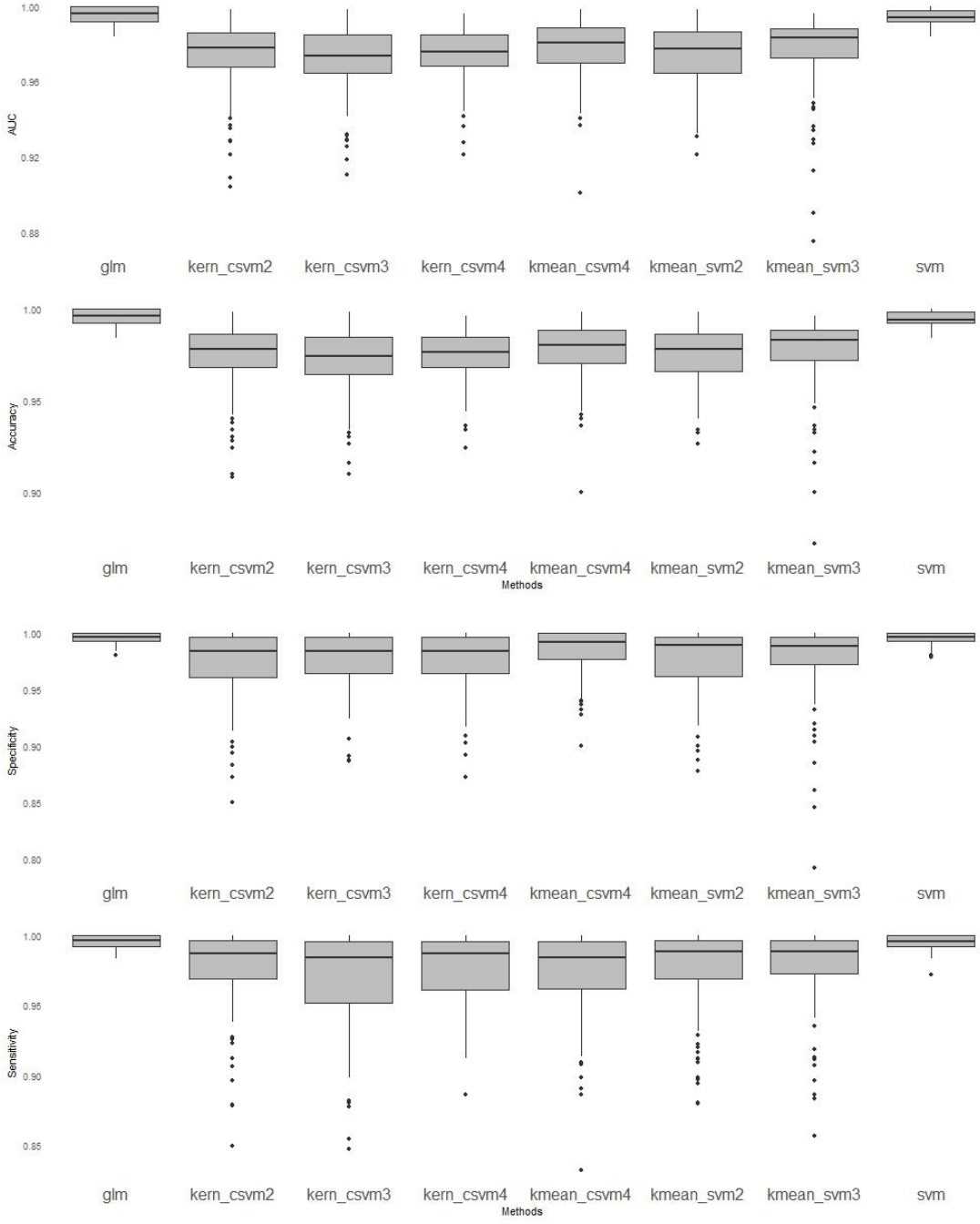
Lampiran 14 : Box Plot Evaluasi Model Studi Simulasi Skenario 3



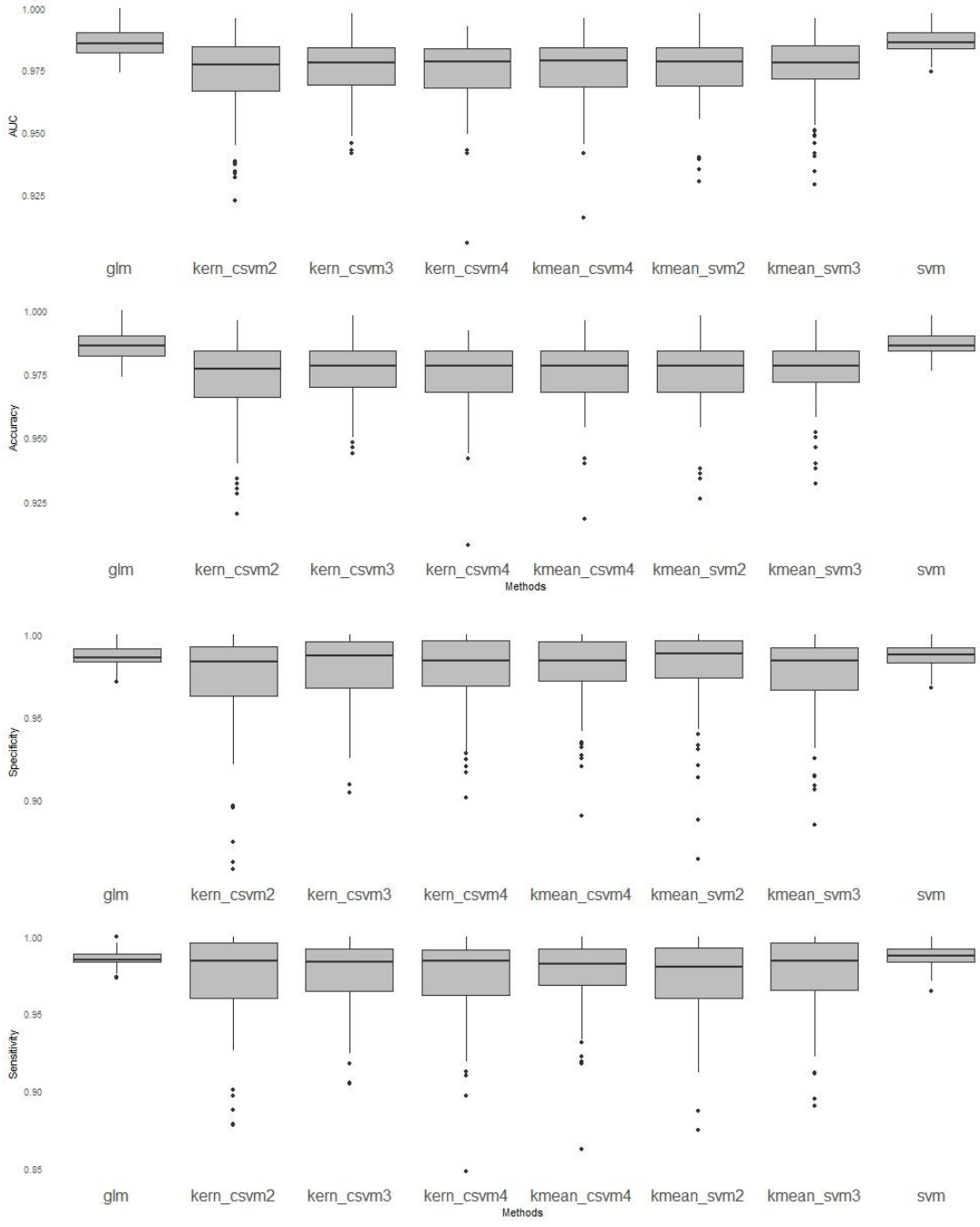
Lampiran 15 : Box Plot Evaluasi Model Studi Simulasi Skenario 4



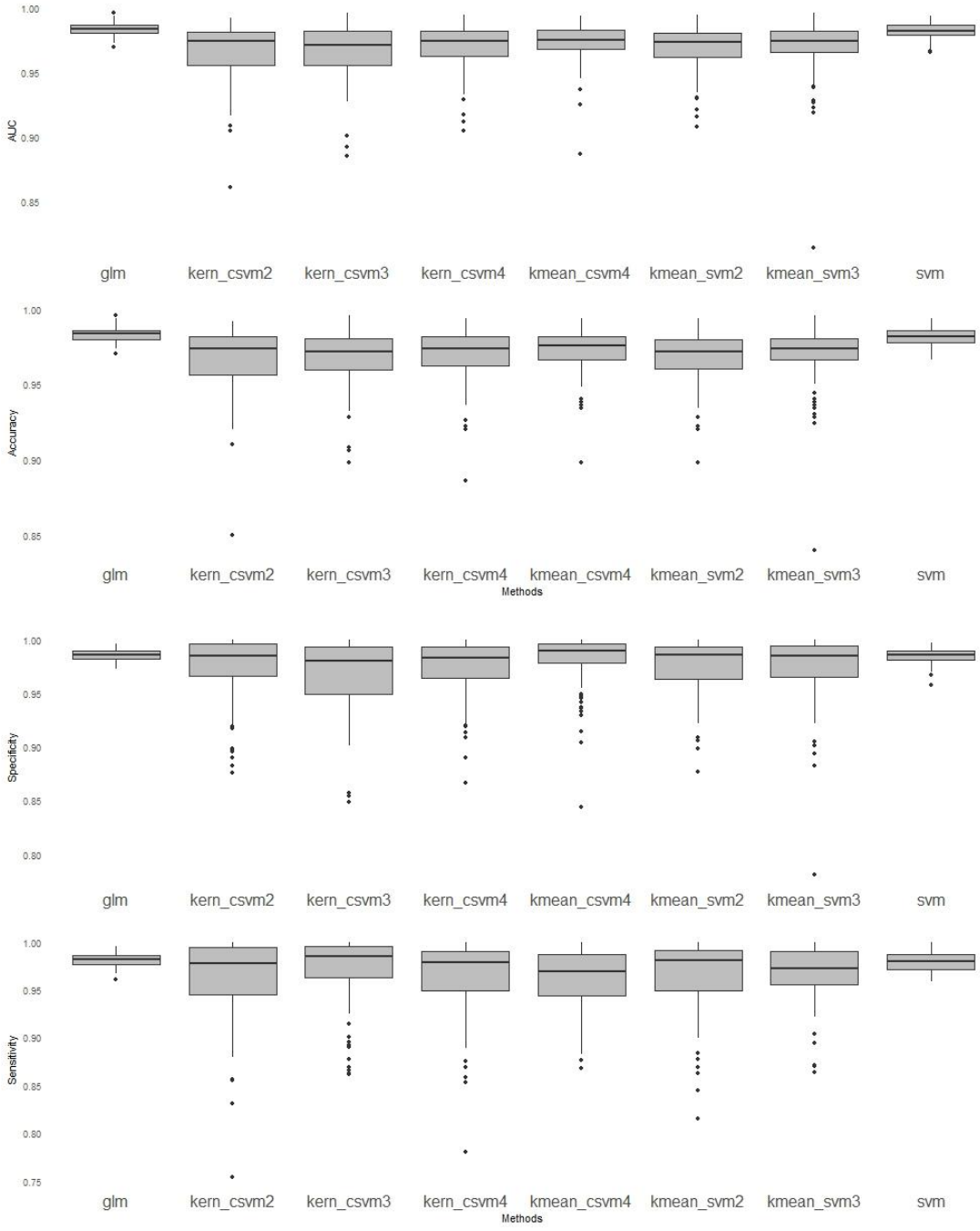
Lampiran 16 : Box Plot Evaluasi Model Studi Simulasi Skenario 5



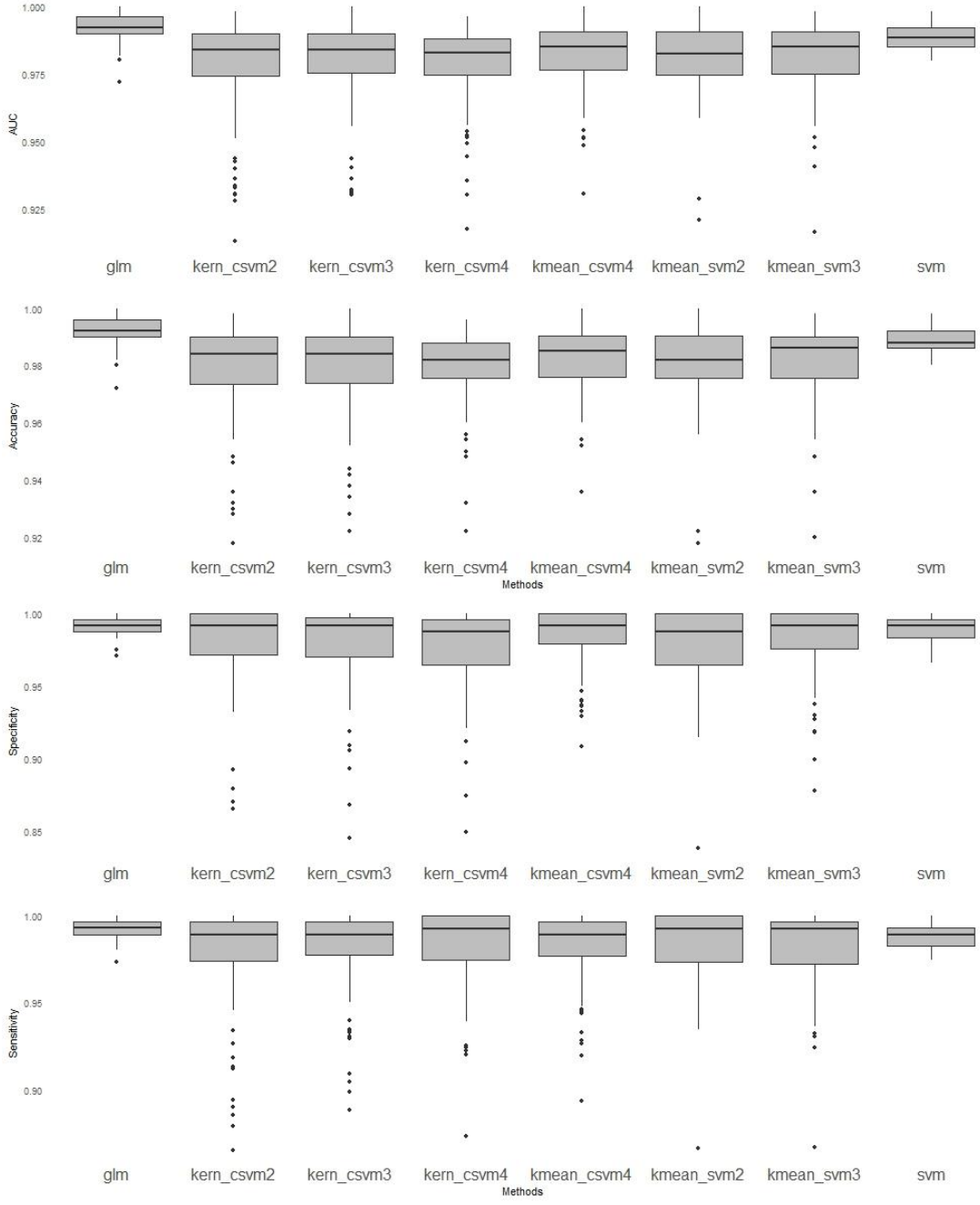
Lampiran 17 : Box Plot Evaluasi Model Studi Simulasi Skenario 6



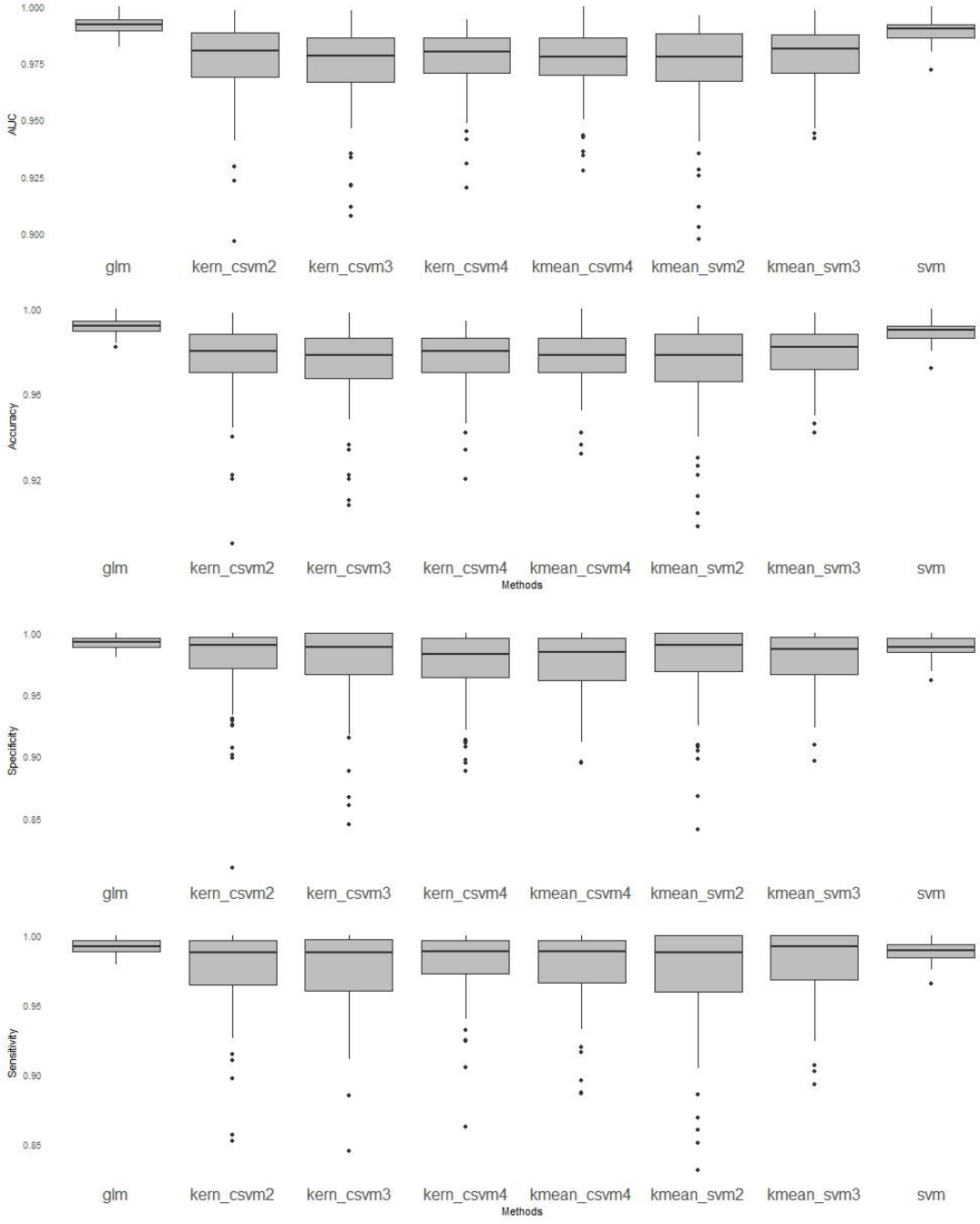
Lampiran 18 : Box Plot Evaluasi Model Studi Simulasi Skenario 7



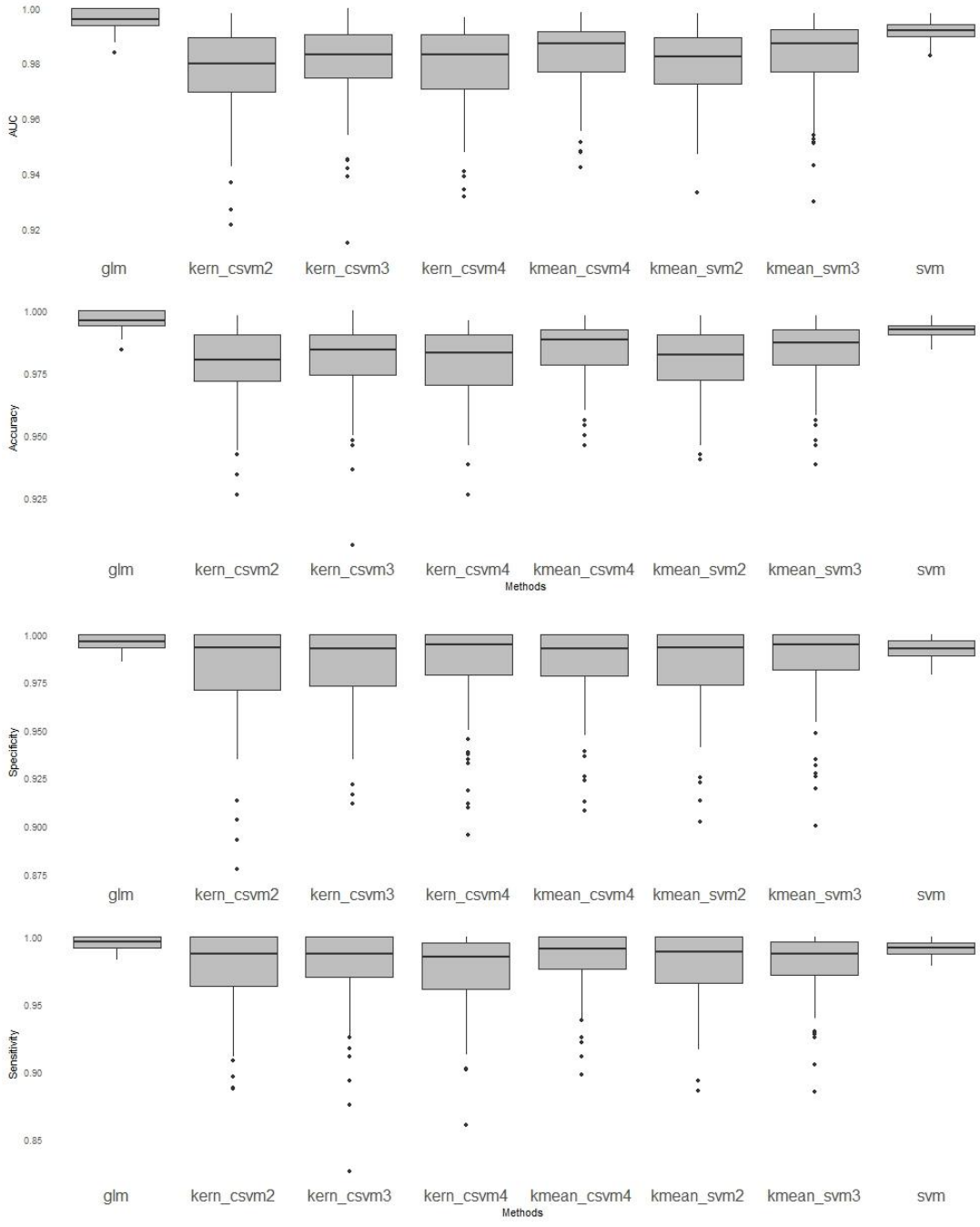
Lampiran 19 : Box Plot Evaluasi Model Studi Simulasi Skenario 8



Lampiran 20 : Box Plot Evaluasi Model Studi Simulasi Skenario 9



Lampiran 21 : Box Plot Evaluasi Model Studi Simulasi Skenario 10



Lampiran 22 : Sintaks R Klasifikasi Data Nasabah Gadai *Fintech* X Beserta Performansinya

```
library(readxl)
library(e1071)
library(MLmetrics)
library(SwarmSVM)

Data_Thesis <- read_excel("D:/Perkuliahan/S2 Statistika/Thesis/Data Thesis.xlsx",sheet = "Sheet4")
Data_train<-list()
Data_test<-list()
for (i in 1:10) {
 set.seed(i)
 sampel <- sample.int(nrow(Data_Thesis), floor(.80*nrow(Data_Thesis)), replace = FALSE)
 Data_train[[i]]<-Data_Thesis[sampel,]
 Data_test[[i]]<-Data_Thesis[-sampel,]
}


Vec_AUC_train<-matrix(ncol=1,nrow=10)
Vec_AUC_test<-matrix(ncol=1,nrow=10)
input_cost_SVM<-seq(10,100, by=10)
input_param_SVM<-seq(1,10,by=1)
comb_param_SVM<-expand.grid(input_cost_SVM,input_param_SVM)
Mat_hasil_train_SVM<-matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))
Mat_hasil_test_SVM<-matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))
for (j in 1:nrow(comb_param_SVM)){
 for (i in 1:10) {
  model_svm<-svm(Y~.,data = Data_train[[i]],type="C-
        classification",cost=comb_param_SVM[j,1],gamma=comb_param_SVM[j,2])
  pred_model_train<-predict(model_svm,Data_train[[i]])
  pred_model_test<-predict(model_svm,Data_test[[i]])
  Vec_AUC_train[i]<-AUC(pred_model_train,Data_train[[i]]$Y)
  Vec_AUC_test[i]<-AUC(pred_model_test,Data_test[[i]]$Y)
 }
 AUC_train<-mean(Vec_AUC_train)
 AUC_test<-mean(Vec_AUC_test)
 hasil_train<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_train))
 hasil_test<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_test))
 Mat_hasil_train_SVM[j,]<-hasil_train
 Mat_hasil_test_SVM[j,]<-hasil_test
}
Mat_hasil_train_SVM<-Mat_hasil_train_SVM[order(Mat_hasil_train_SVM[,3],decreasing=TRUE),]
Mat_hasil_test_SVM<-Mat_hasil_test_SVM[order(Mat_hasil_test_SVM[,3],decreasing=TRUE),]
head(Mat_hasil_train_SVM)
head(Mat_hasil_test_SVM)

Vec_AUC_train<-matrix(ncol=1,nrow=10)
Vec_AUC_test<-matrix(ncol=1,nrow=10)
for (i in 1:10) {
 model_glm <- glm(Y~-1+X1+X2+X3+X4+X5+X6,data=Data_train[[i]],family = binomial())
 pred_glm_train <- ifelse(predict(model_glm,Data_train[[i]],type="response")>0.5,1,0)
 pred_glm_test<-ifelse(predict(model_glm,Data_test[[i]],type="response")>0.5,1,0)
 Vec_AUC_train[i,]<-AUC(pred_glm_train,Data_train[[i]]$Y)
 Vec_AUC_test[i,]<-AUC(pred_glm_test,Data_test[[i]]$Y)
}
mean(Vec_AUC_test)
mean(Vec_AUC_train)


#KMEAN DENGAN K=2
Vec_AUC_train<-matrix(ncol=1,nrow=10)
Vec_AUC_test<-matrix(ncol=1,nrow=10)
```

```
Mat_hasil_train_kmean_cSVM2<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))
Mat_hasil_test_kmean_cSVM2<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))

for (j in 1:nrow(comb_param_SVM)){
 for (i in 1:10){
  Modelkmean_csvm2 <- clusterSVM(x = as.matrix(Data_train[[i]][,-7]), y =
        as.matrix(Data_train[[i]][,7]),seed = 20+i, cost=comb_param_SVM[j,1],lambda =
        comb_param_SVM[j,2], centers = 2, verbose = 0, cluster.method="kmeans",valid.x =
        as.matrix(Data_train[[i]][,-7]),valid.y = as.matrix(Data_train[[i]][,7]))
  predkmean_csvm2_train <- unlist(predict(Modelkmean_csvm2, as.matrix(Data_train[[i]][,-7])))
  predkmean_csvm2_test<-unlist(predict(Modelkmean_csvm2,as.matrix(Data_test[[i]][,-7])))
  Vec_AUC_train[i]<-AUC(predkmean_csvm2_train,Data_train[[i]]$Y)
  Vec_AUC_test[i]<-AUC(predkmean_csvm2_test,Data_test[[i]]$Y)
 }
 AUC_train<-mean(Vec_AUC_train)
 AUC_test<-mean(Vec_AUC_test)
 hasil_train<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_train))
 hasil_test<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_test))
 Mat_hasil_train_kmean_cSVM2[j,]<-hasil_train
 Mat_hasil_test_kmean_cSVM2[j,]<-hasil_test
}
Mat_hasil_train_kmean_cSVM2<-
        Mat_hasil_train_kmean_cSVM2[order(Mat_hasil_train_kmean_cSVM2[,3],decreasing=TRUE),]
Mat_hasil_test_kmean_cSVM2<-
        Mat_hasil_test_kmean_cSVM2[order(Mat_hasil_test_kmean_cSVM2[,3],decreasing=TRUE),]
head(Mat_hasil_train_kmean_cSVM2)
head(Mat_hasil_test_kmean_cSVM2)

#KERNEL K MEANS DENGAN K=2
Vec_AUC_train<-matrix(ncol=1,nrow=10)
Vec_AUC_test<-matrix(ncol=1,nrow=10)
Mat_hasil_train_kernkmean_cSVM2<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))
Mat_hasil_test_kernkmean_cSVM2<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))

for (j in 1:nrow(comb_param_SVM)){
 for (i in 1:10){
  Modelkernkmean_csvm2 <- clusterSVM(x = as.matrix(Data_train[[i]][,-7]), y =
        as.matrix(Data_train[[i]][,7]),seed = 30+i, cost=comb_param_SVM[j,1],lambda =
        comb_param_SVM[j,2], centers = 2, verbose = 0, cluster.method="kmeans",valid.x =
        as.matrix(Data_train[[i]][,-7]),valid.y = as.matrix(Data_train[[i]][,7]))
  predkernkmean_csvm2_train <- unlist(predict(Modelkernkmean_csvm2, as.matrix(Data_train[[i]][,-
        7])))
  predkernkmean_csvm2_test<-unlist(predict(Modelkernkmean_csvm2,as.matrix(Data_test[[i]][,-7])))
  Vec_AUC_train[i]<-AUC(predkernkmean_csvm2_train,Data_train[[i]]$Y)
  Vec_AUC_test[i]<-AUC(predkernkmean_csvm2_test,Data_test[[i]]$Y)
 }
 AUC_train<-mean(Vec_AUC_train)
 AUC_test<-mean(Vec_AUC_test)
 hasil_train<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_train))
 hasil_test<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_test))
 Mat_hasil_train_kernkmean_cSVM2[j,]<-hasil_train
 Mat_hasil_test_kernkmean_cSVM2[j,]<-hasil_test
}
Mat_hasil_train_kernkmean_cSVM2<-
        Mat_hasil_train_kernkmean_cSVM2[order(Mat_hasil_train_kernkmean_cSVM2[,3],decreasing=T
        RUE),]
Mat_hasil_test_kernkmean_cSVM2<-
        Mat_hasil_test_kernkmean_cSVM2[order(Mat_hasil_test_kernkmean_cSVM2[,3],decreasing=TR
        UE),]
```

```
head(Mat_hasil_train_kernkmean_cSVM2)
head(Mat_hasil_test_kernkmean_cSVM2)

#KMEAN DENGAN K=3
Vec_AUC_train<-matrix(ncol=1,nrow=10)
Vec_AUC_test<-matrix(ncol=1,nrow=10)
Mat_hasil_train_kmean_cSVM3<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))
Mat_hasil_test_kmean_cSVM3<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))

for (j in 1:nrow(comb_param_SVM)){
 for (i in 1:10){
  Modelkmean_csvm3 <- clusterSVM(x = as.matrix(Data_train[[i]][,-7]), y =
        as.matrix(Data_train[[i]][,7]),seed = 40+i, cost=comb_param_SVM[j,1],lambda =
        comb_param_SVM[j,2], centers = 3, verbose = 0, cluster.method="kmeans",valid.x =
        as.matrix(Data_train[[i]][,-7]),valid.y = as.matrix(Data_train[[i]][,7]))
  predkmean_csvm3_train <- unlist(predict(Modelkmean_csvm3, as.matrix(Data_train[[i]][,-7])))
  predkmean_csvm3_test<-unlist(predict(Modelkmean_csvm3,as.matrix(Data_test[[i]][,-7])))
  Vec_AUC_train[i]<-AUC(predkmean_csvm3_train,Data_train[[i]]$Y)
  Vec_AUC_test[i]<-AUC(predkmean_csvm3_test,Data_test[[i]]$Y)
 }
 AUC_train<-mean(Vec_AUC_train)
 AUC_test<-mean(Vec_AUC_test)
 hasil_train<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_train))
 hasil_test<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_test))
 Mat_hasil_train_kmean_cSVM3[j,]<-hasil_train
 Mat_hasil_test_kmean_cSVM3[j,]<-hasil_test
}
Mat_hasil_train_kmean_cSVM3<-
        Mat_hasil_train_kmean_cSVM3[order(Mat_hasil_train_kmean_cSVM3[,3],decreasing=TRUE),]
Mat_hasil_test_kmean_cSVM3<-
        Mat_hasil_test_kmean_cSVM3[order(Mat_hasil_test_kmean_cSVM3[,3],decreasing=TRUE),]
head(Mat_hasil_train_kmean_cSVM3)
head(Mat_hasil_test_kmean_cSVM3)

#KERNEL K MEANS DENGAN K=3
Vec_AUC_train<-matrix(ncol=1,nrow=10)
Vec_AUC_test<-matrix(ncol=1,nrow=10)
Mat_hasil_train_kernkmean_cSVM3<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))
Mat_hasil_test_kernkmean_cSVM3<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))

for (j in 1:nrow(comb_param_SVM)){
 for (i in 1:10){
  Modelkernkmean_csvm3 <- clusterSVM(x = as.matrix(Data_train[[i]][,-7]), y =
        as.matrix(Data_train[[i]][,7]),seed = 50+i, cost=comb_param_SVM[j,1],lambda =
        comb_param_SVM[j,2], centers = 3, verbose = 0, cluster.method="kmeans",valid.x =
        as.matrix(Data_train[[i]][,-7]),valid.y = as.matrix(Data_train[[i]][,7]))
  predkernkmean_csvm3_train <- unlist(predict(Modelkmean_csvm3, as.matrix(Data_train[[i]][,-7])))
  predkernkmean_csvm3_test<-unlist(predict(Modelkmean_csvm3,as.matrix(Data_test[[i]][,-7])))
  Vec_AUC_train[i]<-AUC(predkernkmean_csvm3_train,Data_train[[i]]$Y)
  Vec_AUC_test[i]<-AUC(predkernkmean_csvm3_test,Data_test[[i]]$Y)
 }
 AUC_train<-mean(Vec_AUC_train)
 AUC_test<-mean(Vec_AUC_test)
 hasil_train<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_train))
 hasil_test<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_test))
 Mat_hasil_train_kernkmean_cSVM3[j,]<-hasil_train
 Mat_hasil_test_kernkmean_cSVM3[j,]<-hasil_test
}
```

```
Mat_hasil_train_kernkmean_cSVM3<-
        Mat_hasil_train_kernkmean_cSVM3[order(Mat_hasil_train_kernkmean_cSVM3[,3],decreasing=T
        RUE),]
Mat_hasil_test_kernkmean_cSVM3<-
        Mat_hasil_test_kernkmean_cSVM3[order(Mat_hasil_test_kernkmean_cSVM3[,3],decreasing=TR
        UE),]
head(Mat_hasil_train_kernkmean_cSVM3)
head(Mat_hasil_test_kernkmean_cSVM3)


#KMEAN DENGAN K=4
Vec_AUC_train<-matrix(ncol=1,nrow=10)
Vec_AUC_test<-matrix(ncol=1,nrow=10)
Mat_hasil_train_kmean_cSVM4<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))
Mat_hasil_test_kmean_cSVM4<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))

for (j in 1:nrow(comb_param_SVM)){
 for (i in 1:10){
  Modelkmean_csvm4 <- clusterSVM(x = as.matrix(Data_train[[i]][,-7]), y =
        as.matrix(Data_train[[i]][,7]),seed = 60+i, cost=comb_param_SVM[j,1],lambda =
        comb_param_SVM[j,2], centers = 4, verbose = 0, cluster.method="kmeans",valid.x =
        as.matrix(Data_train[[i]][,-7]),valid.y = as.matrix(Data_train[[i]][,7]))
  predkmean_csvm4_train <- unlist(predict(Modelkmean_csvm4, as.matrix(Data_train[[i]][,-7])))
  predkmean_csvm4_test<-unlist(predict(Modelkmean_csvm4,as.matrix(Data_test[[i]][,-7])))
  Vec_AUC_train[i]<-AUC(predkmean_csvm4_train,Data_train[[i]]$Y)
  Vec_AUC_test[i]<-AUC(predkmean_csvm4_test,Data_test[[i]]$Y)
 }
 AUC_train<-mean(Vec_AUC_train)
 AUC_test<-mean(Vec_AUC_test)
 hasil_train<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_train))
 hasil_test<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_test))
 Mat_hasil_train_kmean_cSVM4[j,]<-hasil_train
 Mat_hasil_test_kmean_cSVM4[j,]<-hasil_test
}
Mat_hasil_train_kmean_cSVM4<-
        Mat_hasil_train_kmean_cSVM4[order(Mat_hasil_train_kmean_cSVM4[,3],decreasing=TRUE),]
Mat_hasil_test_kmean_cSVM4<-
        Mat_hasil_test_kmean_cSVM4[order(Mat_hasil_test_kmean_cSVM4[,3],decreasing=TRUE),]
head(Mat_hasil_train_kmean_cSVM4)
head(Mat_hasil_test_kmean_cSVM4)

#KERNEL K MEANS DENGAN K=4
Vec_AUC_train<-matrix(ncol=1,nrow=10)
Vec_AUC_test<-matrix(ncol=1,nrow=10)
Mat_hasil_train_kernkmean_cSVM4<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))
Mat_hasil_test_kernkmean_cSVM4<-
        matrix(ncol=ncol(comb_param_SVM)+1,nrow=nrow(comb_param_SVM))

for (j in 1:nrow(comb_param_SVM)){
 for (i in 1:10){
  Modelkernkmean_csvm4 <- clusterSVM(x = as.matrix(Data_train[[i]][,-7]), y =
        as.matrix(Data_train[[i]][,7]),seed = 70+i, cost=comb_param_SVM[j,1],lambda =
        comb_param_SVM[j,2], centers = 4, verbose = 0, cluster.method="kmeans",valid.x =
        as.matrix(Data_train[[i]][,-7]),valid.y = as.matrix(Data_train[[i]][,7]))
  predkernkmean_csvm4_train <- unlist(predict(Modelkmean_csvm4, as.matrix(Data_train[[i]][,-7])))
  predkernkmean_csvm4_test<-unlist(predict(Modelkmean_csvm4,as.matrix(Data_test[[i]][,-7])))
  Vec_AUC_train[i]<-AUC(predkernkmean_csvm4_train,Data_train[[i]]$Y)
  Vec_AUC_test[i]<-AUC(predkernkmean_csvm4_test,Data_test[[i]]$Y)
 }
 AUC_train<-mean(Vec_AUC_train)
```

```
 AUC_test<-mean(Vec_AUC_test)
 hasil_train<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_train))
 hasil_test<-as.numeric(c(comb_param_SVM[j,1],comb_param_SVM[j,2],AUC_test))
 Mat_hasil_train_kernkmean_cSVM4[j,]<-hasil_train
 Mat_hasil_test_kernkmean_cSVM4[j,]<-hasil_test
}
Mat_hasil_train_kernkmean_cSVM4<-
        Mat_hasil_train_kernkmean_cSVM4[order(Mat_hasil_train_kernkmean_cSVM4[,3],decreasing=T
        RUE),]
Mat_hasil_test_kernkmean_cSVM4<-
        Mat_hasil_test_kernkmean_cSVM4[order(Mat_hasil_test_kernkmean_cSVM4[,3],decreasing=TR
        UE),]
head(Mat_hasil_train_kernkmean_cSVM4)
head(Mat_hasil_test_kernkmean_cSVM4)

colnames(Mat_hasil_train_SVM)<-c("Cost","Gamma","AUC")
colnames(Mat_hasil_test_SVM)<-c("Cost","Gamma","AUC")
colnames(Mat_hasil_train_kmean_cSVM2)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_test_kmean_cSVM2)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_train_kernkmean_cSVM2)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_test_kernkmean_cSVM2)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_train_kmean_cSVM3)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_test_kmean_cSVM3)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_train_kernkmean_cSVM3)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_test_kernkmean_cSVM3)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_train_kmean_cSVM4)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_test_kmean_cSVM4)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_train_kernkmean_cSVM4)<-c("Cost","Lambda","AUC")
colnames(Mat_hasil_test_kernkmean_cSVM4)<-c("Cost","Lambda","AUC")

write.csv(Mat_hasil_train_SVM,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Train_SVM.csv",row.names = FALSE)
write.csv(Mat_hasil_test_SVM,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Test_SVM.csv",row.names = FALSE)
write.csv(Mat_hasil_train_kmean_cSVM2,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Train_Kmean2.csv",row.names = FALSE)
write.csv(Mat_hasil_test_kmean_cSVM2,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Test_Kmean2.csv",row.names = FALSE)
write.csv(Mat_hasil_train_kernkmean_cSVM2,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Train_KernKmean2.csv",row.names = FALSE)
write.csv(Mat_hasil_test_kernkmean_cSVM2,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Test_KernKmean2.csv",row.names = FALSE)
write.csv(Mat_hasil_train_kmean_cSVM3,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Train_Kmean3.csv",row.names = FALSE)
write.csv(Mat_hasil_test_kmean_cSVM3,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Test_Kmean3.csv",row.names = FALSE)
write.csv(Mat_hasil_train_kernkmean_cSVM3,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Train_KernKmean3.csv",row.names = FALSE)
write.csv(Mat_hasil_test_kernkmean_cSVM3,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Test_KernKmean3.csv",row.names = FALSE)
write.csv(Mat_hasil_train_kmean_cSVM4,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Train_Kmean4.csv",row.names = FALSE)
write.csv(Mat_hasil_test_kmean_cSVM4,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Test_Kmean4.csv",row.names = FALSE)
write.csv(Mat_hasil_train_kernkmean_cSVM4,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Train_KernKmean4.csv",row.names = FALSE)
write.csv(Mat_hasil_test_kernkmean_cSVM4,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Klasifikasi/Test_KernKmean4.csv",row.names = FALSE)
```

Lampiran 23 : Sintaks R Analisis Survival dengan *Cox Proportional Hazard* Data Nasabah Gadai *Fintech* X Beserta Performansinya

```
library(readxl)
library(survival)
library(survminer)
library(ggplot2)
library(ggpubr)

#Data Nasabah Early Payment
Dataset0 <- read_excel("D:/Perkuliahan/S2 Statistika/Thesis/Data Thesis.xlsx", sheet="Survival0")
Dataset_x0<-Dataset0[,3:8]
Delta<-as.matrix(Dataset0[,2])
Waktu<-as.matrix(Dataset0[,1])
Dataset0 <- within(Dataset0, {
  X4 <- factor(X4, labels=c('0','1'))
  X5 <- factor(X5, labels=c('0','1'))
  X6 <- factor(X6, labels=c('0','1'))
})

#Membuat Kurva Kaplan Meier dan Uji Log Rank
fit1<- survfit(Surv(T, Status, type="right") ~ X4, data=Dataset0)
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Survival/Plot1_0.jpeg",width=600,height = 400,quality=100)
plot1=ggsurvplot(fit1, data = Dataset0,pval = TRUE,legend.labs = c("Perempuan", "Laki-Laki"))
ggpar(plot1,font.legend = list(size = 14,face="bold"))
dev.off()
fit2<- survfit(Surv(T, Status, type="right") ~ X5, data=Dataset0)
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Survival/Plot2_0.jpeg",width=600,height = 400,quality=100)
plot2<-ggsurvplot(fit2, data = Dataset0,pval = TRUE,legend.labs = c("Nasabah Baru", "Nasabah Lama"))
ggpar(plot2,font.legend = list(size = 14,face="bold"))
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
        Survival/Plot3_0.jpeg",width=600,height = 400,quality=100)
fit3<- survfit(Surv(T, Status, type="right") ~ X6, data=Dataset0)
plot3<-ggsurvplot(fit3, data = Dataset0,pval = TRUE,legend.labs = c("Tidak", "Ya"))
ggpar(plot3,font.legend = list(size = 14,face="bold"))
dev.off()

CoxModel.0 <- coxph(Surv(T, Status, type="right") ~ X1 + X2 + X3 + X4 +
                X5 + X6, data=Dataset0)
summary(CoxModel.0)
uji_0=cox.zph(CoxModel.0,transform=rank)
prognostik_indeks<-as.matrix(Dataset_x0)%*%CoxModel.0$coef

#CIndeks
memory.limit(size=10000)
library(combinat)
comprog<-combn(prognostik_indeks,2)
comprog<-t(comprog)
prog_i<-cbind(comprog[,1])
prog_j<-cbind(comprog[,2])
selisih_prog<-prog_j-prog_i
comtime<-combn(Waktu,2)
comtime<-t(comtime)
ti<-cbind(comtime[,1])
tj<-cbind(comtime[,2])
selisih_t<-tj-ti
indikator<-selisih_prog*selisih_t
for (i in 1:length(indikator)){
  if(indikator[i]<0) {indikator[i]=0} else {indikator[i]=1}
}
```

```
comstatus<-combn(Delta,2)
comstatus<-t(comstatus)
status_i=cbind(comstatus[,1])
v<-matrix(0,nrow=length(selisih_t),ncol=1)
for (i in 1 : length(selisih_t)){
  if((selisih_t[i]>0)&status_i[i]==1) {v[i]=1} else {v[i]=0}
}
cindeks=sum(indikator*v)/sum(v)
cindeks

#Data Nasabah Late Payment
Dataset1 <- read_excel("D:/Perkuliahan/S2 Statistika/Thesis/Data Thesis.xlsx", sheet="Survival1")
Dataset_x1<-Dataset1[,3:8]
Delta<-as.matrix(Dataset1[,2])
Waktu<-as.matrix(Dataset1[,1])
Dataset1 <- within(Dataset1, {
  X4 <- factor(X4, labels=c('0','1'))
  X5 <- factor(X5, labels=c('0','1'))
  X6 <- factor(X6, labels=c('0','1'))
})
fit4<- survfit(Surv(T, Status, type="right") ~ X4, data=Dataset1)
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
          Survival/Plot1_1.jpeg",width=600,height = 400,quality=100)
plot4=ggsurvplot(fit4, data = Dataset1,pval = TRUE,legend.labs = c("Perempuan", "Laki-Laki"))
ggpar(plot4,font.legend = list(size = 14,face="bold"))
dev.off()
fit5<- survfit(Surv(T, Status, type="right") ~ X5, data=Dataset1)
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
          Survival/Plot2_1.jpeg",width=600,height = 400,quality=100)
plot5<-ggsurvplot(fit5, data = Dataset1,pval = TRUE,legend.labs = c("Nasabah Baru", "Nasabah Lama"))
ggpar(plot5,font.legend = list(size = 14,face="bold"))
dev.off()
jpeg(file="D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
          Survival/Plot3_1.jpeg",width=600,height = 400,quality=100)
fit6<- survfit(Surv(T, Status, type="right") ~ X6, data=Dataset1)
plot6<-ggsurvplot(fit6, data = Dataset1,pval = TRUE,legend.labs = c("Tidak", "Ya"))
ggpar(plot6,font.legend = list(size = 14,face="bold"))
dev.off()
CoxModel.1 <- coxph(Surv(T, Status, type="right") ~ X1 + X2 + X3 + X4 +
                X5 + X6, method="efron", data=Dataset1)
summary(CoxModel.1)
uji_1=cox.zph(CoxModel.1,transform=rank)
prognostik_indeks<-as.matrix(Dataset_x1)%*%CoxModel.1$coef

#CIndeks
memory.limit(size=10000)
library(combinat)
comprog<-combn(prognostik_indeks,2)
comprog<-t(comprog)
prog_i<-cbind(comprog[,1])
prog_j<-cbind(comprog[,2])
selisih_prog<-prog_j-prog_i
comtime<-combn(Waktu,2)
comtime<-t(comtime)
ti<-cbind(comtime[,1])
tj<-cbind(comtime[,2])
selisih_t<-tj-ti
indikator<-selisih_prog*selisih_t
for (i in 1:length(indikator)){
  if(indikator[i]<0) {indikator[i]=0} else {indikator[i]=1}
}
comstatus<-combn(Delta,2)
comstatus<-t(comstatus)
```

```
status_i=cbind(comstatus[,1])
v<-matrix(0,nrow=length(selisih_t),ncol=1)
for (i in 1 : length(selisih_t)){
  if((selisih_t[i]>0)&status_i[i]==1) {v[i]=1} else {v[i]=0}
}
cindeks=sum(indikator*v)/sum(v)
cindeks
```

Lampiran 24 : Sintaks R Analisis Survival dengan *Survival* SVM Data Nasabah Gadai *Fintech* X Beserta Performansinya

```
library(readxl)
library(survivalsvm)
#indeks 0 menunjukkan untuk early payment sedangkan 1 untuk late payment
Dataset0 <- read_excel("D:/Perkuliahan/S2 Statistika/Thesis/Data Thesis.xlsx", sheet="Survival0")
Dataset1 <- read_excel("D:/Perkuliahan/S2 Statistika/Thesis/Data Thesis.xlsx", sheet="Survival1")
Model_surv0<-survivalsvm(Surv(T, Status) ~ ., Dataset0, type="hybrid",diff.meth="makediff2",kernel =
"rbf_kernel",kernel.pars =1, gamma.mu=c(0.1,0.3))
Model_surv1<-survivalsvm(Surv(T, Status) ~ ., Dataset1, type="hybrid",diff.meth="makediff2",kernel =
"rbf_kernel",kernel.pars = 1 gamma.mu=c(0.1,0.3))
Hasil_surv0<-predict(Model_surv0,Dataset0)
Hasil_surv1<-predict(Model_surv1,Dataset1)

#Mendapatkan Prognostik Indeks
prog_surv0<-c(Hasil_surv0$predicted)
prog_surv1<-c(Hasil_surv1$predicted)
write.csv(prog_surv0,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Survival/Prog_SVM0.csv",row.names = FALSE)
write.csv(prog_surv1,"D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Survival/Prog_SVM1.csv",row.names = FALSE)

#menghitung C indeks untuk early payment
prognostik_indeks<-read.csv("D:/Perkuliahan/S2 Statistika/Thesis/PROPOSAL/Hasil
Survival/Prog_SVM0.csv")
prognostik_indeks<-as.matrix(prognostik_indeks)
Delta<-as.matrix(Dataset1[,2])
Waktu<-as.matrix(Dataset1[,1])
library(combinat)
comprog<-combn(prognostik_indeks,2)
comprog<-t(comprog)
prog_i<-cbind(comprog[,1])
prog_j<-cbind(comprog[,2])
selisih_prog<-prog_j-prog_i
comtime<-combn(Waktu,2)
comtime<-t(comtime)
ti<-cbind(comtime[,1])
tj<-cbind(comtime[,2])
selisih_t<-tj-ti
indikator<-selisih_prog*selisih_t
for (i in 1:length(indikator)){
  if(indikator[i]<0) {indikator[i]=0} else {indikator[i]=1}
}
comstatus<-combn(Delta,2)
comstatus<-t(comstatus)
status_i=cbind(comstatus[,1])
v<-matrix(0,nrow=length(selisih_t),ncol=1)
for (i in 1 : length(selisih_t)){
  if((selisih_t[i]>0)&status_i[i]==1) {v[i]=1} else {v[i]=0}
}
cindeks=sum(indikator*v)/sum(v)
cindeks
```

BIOGRAFI PENULIS

Nama lengkap penulis adalah Mohammad Alfan Alfian Riyadi, biasa dipanggil Alfan. Penulis lahir di Kota Surabaya tanggal 22 Maret 1994. Penulis merupakan anak tunggal dari pasangan Fajar Riyadi dan Enny Halimah Sadiyah. Hobi yang digeluti penulis adalah membaca dan bermain bulu tangkis. Semasa kuliah sarjana, kegiatan organisasi penulis adalah sebagai staf Analisis Data PSt HIMASTA-ITS dan staf Departemen Syiar di lembaga dakwah jurusan FORSIS-ITS. Kemudian di tahun ketiga mendapatkan amanah sebagai Ketua Departemen Syiar FORSIS-ITS. Penulis juga pernah menjadi asisten dosen untuk beberapa mata kuliah yakni Komputasi Statistika, Pengantar Metode Statistika dan Biostatistika. Pernah mengikuti beberapa pelatihan dan bertemu dengan mahasiswa seluruh Indonesia adalah hal yang paling berkesan selama kuliah. Diantaranya, menjadi peserta President Youth Leadership Camp, dan PGN Innovation Camp. Penulis bersama timnya juga pernah berkesempatan menjadi juara 3 di Lomba Karya Tulis Ilmiah yang diselenggarakan Universitas Islam Indonesia Yogyakarta. Kemudian penulis mendapatkan kesempatan melanjutkan studi S2 dengan Beasiswa Fresh Graduate. Semasa kuliah S2 penulis aktif dalam organisasi Data Science Indonesia Chapter East Java sebagai PIC Internal. Penulis juga beberapa kali mengikuti international conference guna mengembangkan kompetensi. Jika ingin berdiskusi lebih lanjut dapat menghubungi: *alfanstatistika@gmail.com*