# DIAGNOSIS PENDERITA PENYAKIT KANKER PARU MENGGUNAKAN *SUPPORT VECTOR MACHINE* DAN *NAÏVE BAYES*

**SKRIPSI**

Oleh

**Muhammad Iqbal Yunan Helmi**
**NIM 161810101019**

**JURUSAN MATEMATIKA**
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**
**UNIVERSITAS JEMBER**
**2021**

# DIAGNOSIS PENDERITA PENYAKIT KANKER PARU MENGGUNAKAN *SUPPORT VECTOR MACHINE* DAN *NAÏVE BAYES*

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

**Muhammad Iqbal Yunan Helmi**
**NIM 161810101019**

**JURUSAN MATEMATIKA**
**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**
**UNIVERSITAS JEMBER**
**2021**

## PERSEMBAHAN

Puji syukur dengan menyebut nama Allah SWT yang Maha Pengasih dan Maha Penyayang dan sholawat serta salam senantiasa tercurahkan kepada junjungan Nabi Muhammad SAW sehingga terselesaikanlah skripsi ini dan saya persembahkan untuk:

1. Keluarga saya yang tercinta, Ibu Purmini, Bapak Yatiran, dan Adik saya Muhammad Wisnu Hanan Asykuro, Mbah Putri Asiyah, Mbah Kakung Muhtadi, Mbok Jami' dan Mbah Kakung Toyono serta seluruh keluarga yang telah mendukung dan memberikan do'a, kasih sayang dan motivasi yang selalu menguatkan di setiap perjalanan hidup saya,
2. Seluruh jajaran guru dan dosen dari TK Al-Hidayah 1 Sumbermulyo, SDN 1 Sumbermulyo, SMPN 1 Siliragung, SMAN 1 Pesanggaran, dan Jurusan Matematika FMIPA Universitas Jember,
3. Teman-teman MISDIRECTION, teman-teman kost East Mastrip 110, dan semua pihak yang selama ini mendukung saya sehingga skripsi ini bisa terselesaikan,
4. UKM Spora dan Himatika "Geokompstat" yang telah memberikan banyak pengalaman organisasi dan juga mengajarkan berhubungan dengan pihak luar kepada saya,
5. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember, SMA Negeri 1 Pesanggaran, SMP Negeri 1 Siliragung, SDN 1 Sumbermulyo dan TK Al-Hidayah 1 Sumbermulyo.

**MOTTO**

Bertindaklah! Seberapa hebat visi Anda dan seberapa bagusnya perencanaan Anda, akan sia-sia jika Anda tidak bertindak.*)

*)Anthony Dio Martin. 2015. https://www.anthonydiomartin.com/ [diakses pada 3 Januari 2021]

**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Iqbal Yunan Helmi

NIM : 161810101019

menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Diagnosis Penderita Penyakit Kanker Paru Menggunakan *Support Vector Machine* dan *Naive Bayes*" adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi. Demikian pernyataan ini saya buat dengan sebenar-benarnya tanpa ada tekanan dan paksaan dari pihak manapun dan bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, Januari 2021

Yang menyatakan,

Muhammad Iqbal Yunan Helmi

NIM 161810101019

**SKRIPSI**

# DIAGNOSIS PENDERITA PENYAKIT KANKER PARU MENGGUNAKAN *SUPPORT VECTOR MACHINE* DAN *NAÏVE BAYES*

Oleh

**Muhammad Iqbal Yunan Helmi**
**NIM 161810101019**

Pembimbing

Dosen Pembimbing Utama : Dian Anggraeni, S.Si., M.Si.
Dosen Pembimbing Anggota : Dr. Alfian Futuhul Hadi, S.Si., M.Si.

**PENGESAHAN**

Skripsi berjudul "Diagnosis Penderita Penyakit Kanker Paru Menggunakan *Support Vector Machine* dan *Naive Bayes*" telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

| | |
|---|---|
| Ketua, | Anggota I, |
| Dian Anggraeni, S.Si., M.Si. | Dr. Alfian Futuhul Hadi, S.Si., M.Si. |
| NIP 198202162006042002 | NIP 197407192000121001 |
| Anggota II, | Anggota III, |
| Dr. Yuliani Setia Dewi,S.Si., M.Si. | Prof. Drs. I Made Tirta, M.Sc., Ph.D. |
| NIP 197407162000032001 | NIP 195912201985031002 |

Mengesahkan

Dekan,

Drs. Achmad Sjaifullah, M.Sc., Ph.D.

NIP 195910091986021001

# RINGKASAN

**Diagnosisi Penderita Penyakit Kkanker Paru Menggunakan *Support Vector Machine* Dan *Naive Bayes***; Muhammad Iqbal Yunan Helmi; 2021; 43 halaman; Jurusan Matematika Fakulta Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Kanker merupakan masalah paling utama dalam bidang kedokteran, menurut data jenis kanker yang menjadi penyebab kematian terbanyak adalah kanker paru, mencapai 1,7 juta kematian pertahun. Ada beberapa faktor yang dapat mempengaruhi terjangkitnya kanker paru, salah satu penyebabnya yaitu mutasi ekspresi genetika sel paru. Jumlah ekspresi genetika tersebut sangat banyak sehingga dibutuhkan sebuah sistem klasifikasi kanker paru yang dapat mengklasifikasikan antara sel yang beresiko kanker paru dan sel sehat. Beberapa metode yang dapat digunakan untuk pengklasifikasi kanker paru dalam ilmu satatistika antara lain *Naïve Bayes* dan *Support Vector Machine(SVM).*

Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane*. SVM adalah metode learning machine yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *inputspace*. Prinsip dasar SVM adalah *linear classifier,* dan selanjutnya dikembangkan agar dapat bekerja pada kasus *non-linear* dengan memasukkan konsep kernel *trick* pada ruang kerja berdimensi tinggi. Sedangkan *Naïve Bayes* atau disebut juga dengan *Bayesian Classification* merupakan metode pengklasifikasian statistik yang didasarkan pada teorema *bayes* yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas. *Naïve Bayes* handal dalam menangani dataset yang berukuran besar serta dapat menangani data yang tidak relevan.

Penelitian ini akan menggunakan data ekspresi gen *microarray* yang terdiri dari 80 individu dengan 2408 variabel gen kanker paru yang kemudian dibagi menjadi data *training* dan data *testing*. Data tersebut dibagi menjadi 75:25 dengan proporsi sama tiap-tiap kelas klasifikasi. Pengujian data *training* dan data *testing*

dilakukan menggunakan metode SVM dan *Naïve Bayes*. Untuk mendapat model yang optimal digunakan metode *k-fold cross validation* pada SVM dan *Naïve Bayes.*

Hasil pengujian pada data *training* menggunakan SVM menghasilkan tingkat akurasi sebesar 100% untuk fungsi kernel *linear*, 85% untuk fungsi kernel *polynomial*, 100% untuk fungsi kernel *radial*, dan 100% untuk fungsi kernel *sigmoid*. Kemudian dilakukan *tuning* parameter untuk mencari parameter terbaik dan nilai *error* terkecil dari setiap fungsi kernel. *Tuning* parameter yang dilakukan berupa parameter *cost* yang bernilai 0.001, 0.01, 0.1, 1, 10, 100 terhadap setiap fungsi kernel menggunakan metode *5-fold* dan *10-fold cross validation*. Dari hasil pengujian dan proses *tuning* didapatkan bahwa kernel *linear* merupakan fungsi kernel terbaik dibandingkan dengan ketiga kernel lainnya. Kernel *linear* juga memiliki nilai error terkecil pada *5 fold* yang sudah dilakukan dengan *cost* parameternya sebesar 0,001. Pengujian data *testing* model SVM akan digunakan fungsi kernel *linear* dengan metode *5 fold cross validation*.

Hasil pengujian pada data training menggunakan *Naïve Bayes* menghasilkan tingkat akurasi sebesar 98,33%. Kemudian dilakukan pengoptimalan model menggunakan metode *5-fold* dan *10-fold cross validation*. Dari hasil pengujian dan proses ini didapatkan bahwa *10-fold cross validation* mampu melakukan klasifikasi yang lebih baik dari pada *5-fold cross validation* ditandai dengan nilai *error* yang lebih kecil. Pengujian data *testing* model *Naïve Bayes* akan digunakan metode *10-fold cross validation.*

Hasil pengujian metode *Naive bayes* dan SVM terhadap data ekspresi genetika kanker paru, dapat diambil kesimpulan bahwa metode SVM memiliki hasil klasifikasi yang lebih baik dari metode *Naïve Bayes*. Pengujian metode SVM memiliki tingkat ketepatan klasifikasi sebesar 90%, sedangkan untuk metode *Naïve Byaes* memiliki ketepatan klasifikasi sebesar 75%. Klasifikasi SVM dari data sel kanker paru menghasilkan 18 data terklasifikasi secara benar dan ada 2 data kesalahan. Klasifikasi *Naïve Bayes* dari sel kanker paru menghasilkan 15 data terklasifikasi secara benar dan ada 5 data kesalahan.

# PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Diagnosis Penderita Penyakit Kanker Paru Menggunakan *Support Vector Machine* dan *Naive Bayes*". Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi mendapatkan dukungan serta bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Dian Anggraeni, S.Si., M.Si. selaku Dosen Pembimbing Utama dan Dr. Alfian Futuhul Hadi, S.Si., M.Si. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, tenaga, pikiran, dan perhatian dalam penulisan skripsi ini;
2. Dr. Yuliani Setia Dewi,S.Si., M.Si. dan Prof. Drs. I Made Tirta, M.Sc., Ph.D. selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun demi kesempurnaan skripsi ini;
3. Ahmad Kamsyakawuni, S.Si., M.Kom. selaku Dosen Pembimbing Akademik yang memberikan berbagai dukungan, motivasi dan pengarahan selama penulis menjadi mahasiswa;
4. Seluruh Dosen dan Staff Karyawan Jurusan Matematika Fakultas MIPA Universitas Jember;
5. Keluarga yang telah memberikan semangat dan doa tulus ikhlas penuh kasih sayangnya;
6. Teman-teman dan semua pihak yang telah membantu dan memberi semangat.

Guna menyempurnakan skripsi ini, penulis menerima kritik dan saran dari berbagai pihak. Penulis berharap, semoga skripsi ini dapat bermanfaat untuk penelitian-penelitian berikutnya.

Jember,
Januari 2021 Penulis

# DAFTAR ISI

Halaman

# DAFTAR TABEL

Halaman

# DAFTAR GAMBAR

Halaman

# BAB 1. PENDAHULUAN

## 1.1 Latar Belakang

Kanker merupakan masalah paling utama dalam bidang kedokteran dan merupakan salah satu dari 10 penyebab kematian utama di dunia serta merupakan penyakit keganasan yang bisa mengakibatkan kematian pada penderitanya karena sel kanker merusak sel lain. Menurut data jenis kanker dari WHO jenis kanker yang menjadi penyebab kematian terbanyak adalah kanker paru, mencapai 1,7 juta kematian pertahun. Disusul kanker lambung (mencapai lebih dari 1 juta kematian pertahun), kanker hati (sekitar 662.000 kematian pertahun), kanker usus besar (655.000 kematian pertahun), dan yang terakhir yaitu kanker payudara (502.000 kematian pertahun).

Kanker paru adalah salah satu jenis penyakit paru yang memerlukan penanganan dan tindakan yang cepat dan terarah. Pengobatan penyakit ini sangat bergantung pada kecekatan ahli paru untuk mendapatkan diagnosis pasti. Diagnosis penyakit ini membutuhkan ketrampilan dan sarana yang tidak sederhana dan memerlukan pendekatan multidisiplin kedokteran. Ada beberapa faktor yang dapat mempengaruhi terjangkitnya kanker paru, salah satu penyebabnya yaitu mutasi ekspresi genetika sel paru. Jumlah ekspresi genetika tersebut sangat banyak sehingga dibutuhkan sebuah sistem klasifikasi kanker paru yang dapat mengklasifikasikan antara sel yang beresiko kanker paru dan sel sehat. Beberapa metode yang dapat digunakan untuk pengklasifikasi kanker paru dalam ilmu satatistika antara lain *Naïve Bayes* dan *Support Vector Machine(SVM).*

Metode *Naive Bayes* merupakan metode yang hanya membutuhkan jumlah data *training* kecil untuk menentukan estimasi paremeter dalam proses pengklasifikasian. Metode ini digunakan untuk menyelesaikan masalah diagnosa dan prediksi. *Naïve Bayes* merupakan teknik prediksi berbasis probabilitas sederhana yang berdasarkan pada model fitur independen, sedangkan klasifikasi menggunakan SVM dapat dijelaskan secara sederhana yaitu usaha untuk mendapatkan garis sebagai fungsi pemisah terbaik yang dapat memisahkan dua kelas yang berbeda pada ruang input. SVM adalah salah satu teknik yang relatif

baru dibandingkan dengan teknik lain, tetapi memiliki performansi yang lebih baik di berbagai bidang aplikasi seperti *bioinformatics*, pengenalan tulisan tangan, klasifikasi teks dan lain sebagainya. Teori yang mendasari SVM sendiri sudah berkembang sejak 1960-an, tetapi baru diperkenalkan oleh Vapnik, Boser dan Guyon pada tahun 1992 dan sejak itu SVM berkembang dengan pesat.

Penelitian Munawarah (2016) telah membuktikan bahwa metode *Support Vector Machine* (SVM) dapat melakukan diagnosis hepatitis berdasarkan data tes fungsi hati. Penelitian sebelumnnya oleh Ayu dan Santi (2012) melakukan diagnosis kanker payudara dengan menggunakan SVM. Hasil penelitian tersebut menunjukkan ketepatan klasifikasi metode SVM mencapai 94%. Hidayatul (2018) membahas mengenai diagnosis penyakit jantung menggunakan metode *Naïve Bayes* dengan ketepatan klasifikasi sebesar 92,31%.

*Naive Bayes* dan SVM merupakan metode algoritma data mining yang digunakan untuk melakukan klasifikasi. Beberapa penelitian yang sebelumnya telah dilakukan, SVM dan *Naïve Bayes* diketahui memiliki akurasai yang tinggi. Penelitian ini dilakukan dengan tujuan untuk melakukan perbandingan antara algoritma *Naive Bayes* dengan SVM dalam memprediksi keberhasilan metode pengobatan kanker paru, karena pengobatan penyakit ini sangat bergantung pada diagnosis pasti.

## 1.2 Rumusan Masalah

Berdasarkan penjelasan pada latar belakang dapat ditarik rumusan masalah sebagai berikut:

1. Bagaimana model klasifikasi penyakit kanker paru menggunakan metode *Supprot Vector Machine* (SVM) ?
2. Bagaimana model klasifikasi penyakit kanker paru menggunakan metode *Naïve Bayes* ?
3. Bagaimana perbandingan hasil klasifikasi penyakit kanker paru model *Support Vector Machine* (SVM) dan model *Naïve Bayes* ?

### 1.3 Tujuan Penelitian

Tujuan dari penelitian ini anatara lain adalah sebagai berikut :

1. Mendapatkan model klasifikasi penyakit kanker paru menggunakan metode *Support Vector Machine* (SVM)
2. Mendapatkan model klasifikasi penyakit kanker paru menggunakan *Naïve Bayes*
3. Mendapatkan hasil perbandingan klasifikasi metode *Support Vector Machine* (SVM) dan metode *Naïve Bayes*

### 1.4 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah memberikan informasi terkait hasil klasifikasi data menggunakan metode *Support Vector Machine* (SVM) dan *Naïve Bayes* dalam diagnosis penyakit kanker paru.

# BAB 2. TINJAUAN PUSTAKA

## 2.1 Kanker Paru-Paru

Kanker paru dalam arti luas adalah semua penyakit keganasan di paru, mencakup keganasan yang berasal dari paru sendiri maupun keganasan dari luar paru (*metastasis* tumor di paru). Kanker paru adalah tumor ganas yang berasal dari epitel bronkus atau karsinoma bronkus (*bronchogenic carcinoma*). Menurut konsep masa kini kanker adalah penyakit gen. Sebuah sel normal dapat menjadi sel kanker apabila oleh berbagai sebab terjadi ketidak seimbangan antara fungsi *onkogen* dengan gen tumor *suppresor* dalam proses tumbuh dan kembangnya sebuah sel. Sel kanker adalah sel normal yang mengalami mutasi/perubahan genetik dan tumbuh tanpa terkoordinasi dengan sel-sel tubuh lain. Perubahan ini berjalan dalam beberapa tahap atau yang dikenal dengan proses *multistep carcinogenesis*. (Jusuf, 2005).

Gambaran klinik penyakit kanker paru tidak banyak berbeda dari penyakit paru lainnya, terdiri dari keluhan subyektif dan gejala obyektif. Dari *anamnesis* akan didapat keluhan utama dan perjalanan penyakit, serta faktor–faktor lain yang sering sangat membantu tegaknya diagnosis. Keluhan utama dapat berupa :batuk-batuk dengan / tanpa dahak (dahak putih, dapat juga purulen), batuk darah, sesak napas, suara serak, sakit dada, sulit / sakit menelan, benjolan di pangkal leher, sembab muka dan leher, kadang-kadang disertai sembab lengan dengan rasa nyeri yang hebat (Kemenkes, 2008). Tidak jarang yang pertama terlihat adalah gejala atau keluhan akibat *metastasis* di luar paru, seperti kelainan yang timbul karena kompresi hebat di otak, pembesaran hepar atau patah tulang kaki. Gejala dan keluhan yang tidak khas seperti berat badan berkurang, nafsu makan hilang, demam hilang timbul, sindrom *paraneoplastik*, seperti "*hypertrophic pulmonary osteoartheopathy*", *trombosis vena perifer* dan *neuropatia* (Jusuf, 2005).

## 2.2 Klasifikasi

Klasifikasi adalah proses penemuan model (atau fungsi) yang biasa digunakan untuk menggambarkan dan membedakan kelas data atau konsep yang bertujuan dapat digunakan untuk prediksi kelas dari objek yang label kelasnya tidak diketahui. Algoritma klasifikasi yang banyak digunakan secara luas, yaitu *Decision/classification trees, Bayesian classifiers/ Naïve Bayes classifiers, Neural networks*, Analisa Statistik, Algoritma Genetika, *Rough sets*, *k-nearest neighbor*, Metode *Rule Based*, *Memory based reasoning*, dan *Support Vector Machines* (SVM) (Han, 2006). Salah satu metode yang dapat digunakan untuk mengukur akurasi algoritma klasifikasi adalah metode *confusion matrix.* Metode ini menggunakan tabel matriks seperti pada Tabel 2.1 jika data set hanya terdiri dari dua kelas, kelas yang satu dianggap sebagai positif dan yang lainnya negatif (Bramer, 2007).

Tabel 2.1 *Confussion Matrix*

| Kelas Asli | Kelas Prediksi | |
|---|---|---|
| | Negatif | Positif |
| Negatif | *True Negative* (TN) | *False Positive* (FP) |
| Positif | *False Negative* (FN) | *True Positive* (TP) |

*True positives* adalah jumlah *record* positif yang diklasifikasikan sebagai positif *false positives* adalah jumlah *record* negatif yang diklasifikasikan sebagai positif, *false negatives* adalah jumlah *record* positif yang diklasifikasikan sebagai negatif, *true negatives* adalah jumlah *record* negatif yang diklasifikasikan sebagai negative.

Berdasarkan tabel 2.1 tersebut dapat diperoleh performa klasifikasi antara lain *accuracy*, *precision*, *sensitivity*, dan *specificity*. Nilai *accuracy* merupakan nilai seberapa besar hasil keakuratan dari klasifikasi data tersebut. *Precision* merupakan rasio perbandingan antara kelas benar positif dengan semua kelas hasil positif. *Sensitifity* merupakan proporsi kelas positif diidentifikasi benar, sedangkan *specitifity* adalah proporsi kelas negatif yang diidentifikasi benar. Berikut formulasi untuk menghitung *accuracy*, *precision*, *sensitivity*, dan *specificity* ditunjukan pada persamaan (2.1), persamaan (2.2), persamaan (2.3), persamaan (2.4) (Han, 2006).

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FN} * 100\% \quad (2.1)$$

$$Precision = \frac{TP}{TP+FP} * 100\% \quad (2.2)$$

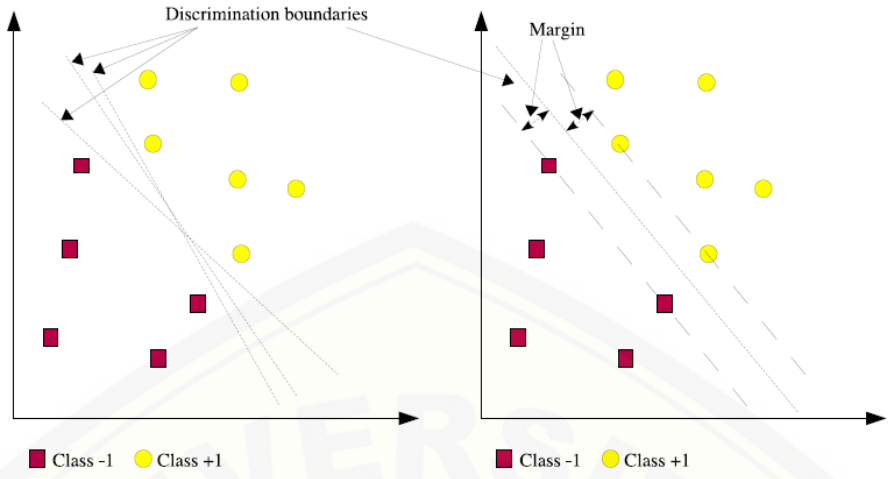
$$Sensitivity = \frac{TP}{TP+FN} * 100\% \quad (2.3)$$

$$Specitifity = \frac{TN}{TN+FP} * 100\% \quad (2.4)$$

### *2.3 Support Vector Machine* (SVM)

*Support Vector Machine* (SVM) dikembangkan oleh Boser, Guyon, Vapnik, dan pertama kali dipresentasikan pada tahun 1992 di *Annual Workshop on Computational Learning Theory*. Konsep dasar SVM sebenarnya merupakan kombinasi harmonis dari teori-teori komputasi yang telah ada puluhan tahun sebelumnya, seperti *margin hyperplane*. SVM adalah metode learning machine yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *inputspace*. Prinsip dasar SVM adalah *linear classifier,* dan selanjutnya dikembangkan agar dapat bekerja pada kasus *non-linear* dengan memasukkan konsep kernel trick pada ruang kerja berdimensi tinggi. (Gun, 1998).

Konsep SVM secara sederhana adalah usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah *class* pada *input space*. Gambar 2.1 memperlihatkan beberapa *pattern* yang merupakan anggota dari dua buah *class* : +1 dan –1. *Pattern* yang tergabung pada class –1 disimbolkan dengan warna merah (kotak), sedangkan *pattern* pada class +1, disimbolkan dengan warna kuning (Nugroho dkk.,2003).




l

Gambar 2.1 *Hyperplane* yang memisahkan kedua *class*

Berbagai alternatif garis pemisah (*discrimination boundaries*) ditunjukkan pada Gambar 2.1. *Hyperplane* pemisah terbaik antara kedua *class* dapat ditemukan dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya. *Margin* adalah jarak antara *hyperplane* dengan *pattern* terdekat dari masing-masing *class*. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Garis solid pada Gambar 2.1 sebelah kanan menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua *class*, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah *support vector*. Usaha untuk mencari titik *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM (Cristianini,2000).

Data yang tersedia dinotasikan sebagai $x_i \in R^p$, sedangkan label masing-masing dinotasikan $y_i \in \{-1, +1\}$ untuk $i = 1,2,\dots,n.$ dimana n adalah banyaknya data. *Positive class* dinotasikan sebagai +1, dan *negative class* sebagai −1. Diasumsikan bahwa kedua *class* -1 dan +1 dapat dipisahkan secara sempurna oleh *hyperplane* di *D-dimensional feature space*, yang didefenisikan sebagai berikut:

$$w.x_i + b = 0 \qquad (2.5)$$

Pattern $x_i$ yang tergolong ke dalam *negative class*, dapat dirumuskan sebagai pattern yang memenuhi pertidaksamaan berikut:

$$w.x_i + b \leq -1 \qquad (2.6)$$

sedangkan pattern $x_i$ yang tergolong ke dalam *positive class* yang memenuhi pertidaksamaan berikut:

$$w.x_i + b \geq 1 \quad (2.7)$$

M*argin* terbesar dapat ditemukan dengan memaksimalkan jarak antara *hyperplane* dan *pattern* terdekat, yaitu $1/\|w\|$ ($\|w\|$ adalah norm dari vektor w). Selanjutnya, masalah ini dirumuskan ke dalam *Quadratic Programming* (QP) problem, dengan mencari titik minimal persamaan seperti berikut.

*Minimize*:

$$\|w\|^2 = w^T w \quad (2.8)$$

dengan memperhatikan persamaan berikut:

$$y_i((w.x_i) + b) - 1 \geq 0, \forall_i \quad (2.9)$$

Problem ini dapat dipecahkan dengan berbagai teknik komputasi, di antaranya *Lagrange Multiplier*:

$$L(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i y_i (w.x_i + b - 1) \quad (2.10)$$

$\alpha_i$ adalah *Langrange multiplier* yang bernilai nol atau positif $\alpha_i > 0$ . Nilai optimal dari persamaan (2.10) dapat diperoleh dengan meminimalkan L terhadap *w* dan *b*, dan memaksimalkan L terhadap $\alpha_i$. Dengan memodifikasi persamaan (2.10), memaksimalkan masalah di atas dapat direpresentasikan dalam $\alpha_i$

*Maximize*:

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1} \alpha_i \alpha_j y_i y_j x_i x_j \quad (2.11)$$

*Subject to*:

$$\alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0 \quad (2.12)$$

Hasil dari perhitungan ini menghasilkan banyak nilai $\alpha_i$ positif. Data yang berkorelasi dengan $\alpha_i$ bernilai positif, merupakan *support vectors*, yaitu data yang memiliki jarak terdekat dengan *hyperplane* (Nugroho, 2007).

2.3.1 *Softmargin*

Penjelasan di atas berdasarkan asumsi bahwa kedua belah *class* dapat terpisah secara sempurna oleh hyperplane. Akan tetapi, umumnya dua buah *class* pada *input space* tidak dapat terpisah secara sempurna. Hal ini menyebabkan *constraint* pada persamaan (2.9) tidak dapat terpenuhi, sehingga optimisasi tidak dapat dilakukan. Untuk mengatasi masalah ini, SVM dirumuskan ulang dengan

memperkenalkan teknik *softmargin*. Dalam *softmargin*, persamaan (2.9) dimodifikasi dengan memasukkan *slack* variable $\xi > 0$ sebagai berikut

$$y_i((w.x_i) + b) \geq 1 - \xi, \forall_i \quad (2.13)$$

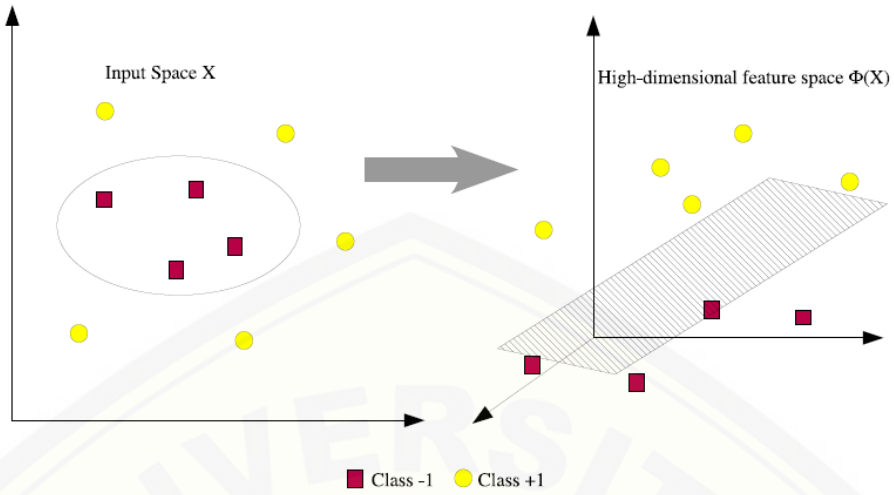
Dengan demikian persamaan (2.8) diubah:

$$\|w\|^2 = w^T w + \mathrm{C} \sum_{i=1}^{n} \xi_i \quad (2.14)$$

Parameter C (*cost*) berfungsi untuk mrngontrol hubungan antara variable *slack* dan margin. Nilai C yang besar berarti akan memberikan penalty yang lebih besar terhadap *error* klasifikasi (Nugroho, 2007).

2.3.2 *Kernel Trick*

Masalah dalam domain dunia nyata tidak semuanya bersifat *linear*, kebanyakan bersifat *non-linear*. Penyelesaikan untuk problem *non-linear* pada SVM, harus melakukan modifikasi SVM dengan memasukkan fungsi Kernel. Modifikasi SVM dalam kasus *non-linear*, pertama-tama data yang berada pada ruang vektor awal ($\{x_i \in \Re^D\}$) berdimensi D, harus dipetakan ke ruang vektor baru yang berdimensi lebih tinggi ($\{x_i' \in \Re^Q\}$). Pada ruang vektor yang baru ini, *hyperplane* yang memisahkan kedua class tersebut dapat dikonstruksikan. Fungsi pemetaan tersebut dinotasikan sebagai $\Phi(x)$. Pemetaan ini bertujuan untuk merepresentasikan data ke dalam format yang *linear separable* pada ruang vektor yang baru. Hal ini sejalan dengan teori Cover yang menyatakan *"Jika suatu transformasi bersifat non linear dan dimensi dari feature space cukup tinggi, maka data pada input space dapat dipetakan ke feature space yang baru, dimana pattern-*Nugroho (2007) mengilustrasi teori Cover yang ditunjukkan pada Gambar 2.2.




Gambar 2.2 Memetakan data ke ruang vektor yang lebih tinggi.

$$\Phi : \Re^D \rightarrow \Re^Q, D < Q \quad (2.15)$$

Proses optimisasi pada fase ini memerlukan perhitungan *dot product* dua buah variabel pada ruang vektor baru. *Dot product* kedua buah vektor $(x_i)$ dan $(x_j)$ dinotasikan sebagai $\Phi(\mathrm{x_i}).\Phi(\mathrm{x_j})$. Nilai *dot product* kedua buah vektor ini dapat dihitung secara tak langsung, yaitu menggunakan *Kernel Trick*. Perumusan dari *Kernel Trick* adalah sebagai berikut:

$$K(x_i, x_j) = \Phi(\mathrm{x}_i).\Phi(x_j) \quad (2.16)$$

*Kernel Trick* memberikan kemudahan dalam menentukan *support vector*. *Support vector* dapat ditemukan tanpa mengetahui bentuk dari fungsi *non-linear* Φ, cukup mengetahui fungsi kernel yang dipakai, dan tidak perlu mengetahui wujud dari fungsi *non-linear* Φ (Nugroho,2003). Berbagai jenis fungsi kernel dikenal, sebagaimana dirangkumkan pada Tabel 2.2.

Tabel 2.2 Fungsi Kernel dalam SVM

| Nama Kernel | Definisi |
|---|---|
| *Polynomial* | $K(x_i, x_j) = (x_i . x_j + 1)^p$ |
| *Radial* | $K(x_i, x_j) = exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$ |
| *Sigmoid* | $K(x_i, x_j) = tanh(\alpha \mathrm{x_i} . \mathrm{x_j} + \beta)$ |
| *Linear* | $K(x_i, x_j) = (x_i, x_j)$ |

## 2.4 *Naïve Bayes*

*Naïve Bayes* yang pertama kali dikemukakan oleh Revered Thomas Bayes. Penggunaan *Naïve Bayes* sudah dikenalkan sejak tahun 1702-1761. *Naïve Bayes* atau disebut juga dengan *Bayesian Classification* merupakan metode pengklasifikasian statistik yang didasarkan pada teorema *bayes* yang dapat digunakan untuk memprediksi probabilitas keanggotaan suatu kelas (Hand, 2001). Sedangkan Kononenko dan Langley menyimpulkan bahwa *Naïve Bayes* merupakan kemungkinan label kelas data atau bisa diasumsikan sebagai atribut kelas yang diberi label (Langley, 1994). *Naïve Bayes* terbukti memiliki akurasi dan kecepatan yang tinggi saat diaplikasikan ke dalam database yang besar (Lewis, 1998). *Naïve Bayes* merupakan salah satu algoritma dalam teknik data mining yang menerapkan teori *bayes* dalam klasifikasi. *Naïve Bayes* handal dalam menangani dataset yang berukuran besar serta dapat menangani data yang tidak relevan (Ridwan, 2013). Simbol untuk X adalah vektor masukan yang berisi data dan Y adalah label kelas. Persaman dari teorema bayes sebagai berikut:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)} \tag{2.17}$$

keterangan:

*X* = Data dengan *class* yang belum diketahui.

*Y* = Hipotesis Data X merupakan suatu *class* spesifik

*P(Y|X)* = probabilitas hipotesis Y berdasarkan kondisi X (*posteriori prob*).

*P(Y)* = Probabilitas hipotesis Y (*prior prob*).

Perlu diketahui bahwa proses klasifikasi *Naïve Bayes* memerlukan sejumlah petunjuk untuk menentukan kelas apa yang cocok bagi sampel yang dianalisis tersebut (Saleh, 2015). Karena itu, metode *Naïve Bayes* diatas disesuaikan sebagai berikut:

$$P(Y_j|X_1, \dots, X_n) = \frac{P(Y)P(X_1,\dots,X_n|Y_j)}{P(X_{1,\dots,}X_n)} \tag{2.18}$$

Dimana variable $Y_j$ mempresentasikan kelas, sementara varaibel $X_1, \dots, X_n$ merepresentasikan karakteristik petunjuk yang dibutuhkan untuk melakukan klasifikasi. Maka rumus tersebut menjelaskan bahwa peluang masuknya sampel karakteristik tertentu dalam kelas $Y_j$ (*Posterior*) adalah peluang munculnya kelas

$Y_j$ (sebelum masuknya sampel tersebut, seringkali disebut *prior*), dikali dengan peluang kemunculan karakteristik-karakteristik sampel pada kelas $Y_j$ (disebut juga *likelihood*), dibagi dengan peluang kemunculan karakteristikkarakteristik sampel secara global (disebut juga evidence). Karena itu, rumus di atas dapat pula ditulis secara sederhana sebagai berikut:

$$Posterior = \frac{prior\ x\ likelihood}{evidence} \tag{2.19}$$

Nilai *evidence* selalu tetap untuk setiap kelas pada satu sampel. Nilai dari *posterior* tersebut nantinya akan dibandingkan dengan nilai-nilai *posterior* kelas lainnya untuk menentukan ke kelas apa suatu sampel akan diklasifikasikan. Penjabaran lebih lanjut rumus *Bayes* tersebut dilakukan dengan menjabarkan $(Y_j|X_1, \dots, X_n)$ menggunakan aturan perkalian sebagai berikut:

$$\begin{aligned} P(Y_j|X_1, \dots, X_n) &= P(Y_j)P(X_1, \dots, X_n|Y_j) \\ &= P(Y_j)P(X_1|Y_j)P(X_2, \dots, X_n|Y_jX_1) \\ &= P(Y_j)P(X_1|Y_j)P(X_2|Y_j, X_1)P(X_3, \dots, X_n|Y_j, X_1X_2) \\ &= P(Y_j)P(X_1|Y_j)P(X_2|Y_j, X_1)P(X_3|Y_j, X_1, X_2)P(X_4, \dots, X_n|Y_j, X_1, X_2, X_3) \\ &= P(Y_j)P(X_1|Y_j)P(X_2|Y_j, X_1)P(X_3|Y_j, X_1, X_2) \dots P(X_n|Y_j, X_1, X_2, X_3, \dots X_{n-1}) \end{aligned} \tag{2.20}$$

Dapat dilihat bahwa hasil penjabaran tersebut menyebabkan semakin kompleksnya faktor-faktor syarat yang mempengaruhi nilai probabilitas, yang hampir mustahil untuk dianalisa satu persatu. Akibatnya, perhitungan tersebut menjadi sulit untuk dilakukan. Disinilah digunakan asumsi independensi yang sangat tinggi (*naive*), bahwa masing-masing petunjuk $(X_1, X_2, \dots, X_n)$ saling bebas (independen) satu sama lain, dengan asumsi tersebut, maka berlaku satu kesamaan sebagai berikut:

$$P(X_i|Y_j) = \frac{P(X_i \cap X_j)}{P(X_j)} = \frac{P(X_i)P(X_j)}{P(X_j)} = P(X_i) \tag{2.21}$$

Untuk $i \neq j$, sehingga

$$P(X_i|Y, X_j) = P(X_i|Y_j) \tag{2.22}$$

Dari persamaan diatas dapat disimpulkan bahwa asumsi independensi *naïve* tersebut membuat syarat peluang menjadi sederhana, sehingga perhitungan menjadi

mungkin untuk dilakukan. Selanjutnya, penjabaran $P(Y_j|X_1,\dots,X_n)$ dapat disederhanakan menjadi:

$$P(X_j|X_1,\dots,X_n) = P(Y_j)P(X_1|Y_j)P(X_2|Y_j)P(X_3|Y_j)\dots P(X_n|Y_j)$$
$$= (P(Y_j)\prod_{i=1}^{n} P(X_i|Y_j)) \quad (2.23)$$

Sehingga hasil klasifikasi merupakan *class* yang menghasilkan nilai probabilitas maksimum atau dapat dinyatakan dalam persamaan sebagai berikut:

$$Y_{MAP} = \arg max_{Y_j \in Y} (P(Y_j)\prod_{i=1}^{n} P(X_i|Y_j)) \quad (2.24)$$

keterangan :

$P(X_j|X_1, \dots,X_n)$ : *Posterior Probability*

$P(X_i|Y_j)$ : *Likelihood*

$P(Y_j)$ : *Prior Probability*

$Y_{MAP}$ : *Class* dengan *Maximum A Posterior Probability*

Persamaan diatas merupakan model dari teorema *Naïve Bayes* yang selanjutnya akan digunakan proses klasifikasi dengan data kuantitatif atau kontinyu digunakan rumus *Densitas Gauss*:

$$P(X_i = x_I|Y = y_j) = \frac{1}{\sqrt{2\pi\sigma_{ij}}} exp^{\frac{(x_i-\mu_{ij})^2}{2\sigma_{ij}^2}} \quad (2.25)$$

dimana :

$P$ = Peluang

$X_i$ = Atribut ke *i*

$x_i$ = Nilai atribut ke *i*

$Y$ = Kelas yang dicari

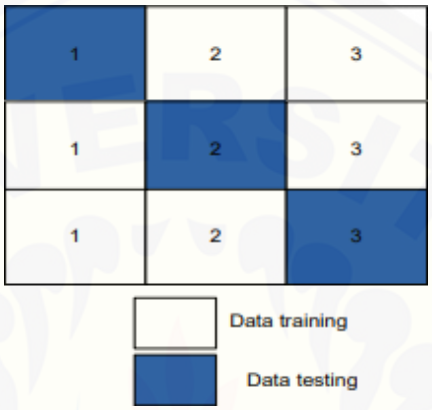
$y_j$ = Sub kelas *Y* yang dicari

$\mu_{ij}$ = *Mean* sampel dari data *training* yang menjadi milik $y_j$

## 2.5 *K-fold Cross Validation*

*K-fold cross validation* merupakan suatu metode digunakan untuk melakukan prediksi model dan memperkirakan seberapa akurat sebuah model prediktif ketika dijalankan dalam praktiknya. Salah satu teknik dari *cross validation* adalah *k-fold cross validation*, yang mana memecah data menjadi k bagian set data

dengan ukuran yang sama. Penggunaan *k-fold cross validation* untuk menghilangkan bias pada data, *training* dan *testing* pada data dilakukan sebanyak k kali, pada percobaan pertama, subset S1 diperlakukan sebagai data pengujian dan subset lainnya diperlakukan sebagai data pelatihan, pada percobaan kedua subset S1, S3,…Sk menjadi data pelatihan dan S2 menjadi data pengujian, dan seterusnya (Bramer, 2007)




Gambar 2.3 Model *3-fold cross validation*

Gambar 2.3 merupakan penggunaan *3-fold cross validation*. Dimana setiap data akan di eksekusi sebanyak 3 kali dan setiap *subset* data akan mempunyai kesempatan sebagai data *testing* atau data *training*. Model pengujian seperti berikut dengan diasumsikan nama setiap pembagian data yaitu D1, D2, dan D3:

1. Percobaan pertama data D1 sebagai data *testing* sedangkan D2 dan D3 sebagai data *training*.
2. Percobaan kedua data D2 sebagai data *testing* sedangkan data D1 dan D3 sebagai data *training*.
3. Pada percobaan terakhir atau percobaan ketiga data D3 sebagai data *testing* sedangkan D1 dan D2 sebagai data *training*.

## 2.6 *Support Vector Machine* dan *Naïve Bayes* pada R

Pada program R terdapat beberapa paket yang dapat digunakan untuk memudahkan dalam mengolah data yang diinginkan. Salah satu paket didalam R yang digunakan dalam penelitian ini yaitu paket SVM dan *Naïve Bayes*. Terdapat empat paket terkait SVM yang ada pada program R yaitu paket `e1071`, `kernlab`,

`klaR`, dan `svmpath`. Paket `e1071` merupakan paket SVM yang berfungsi untuk metode penyelesaian parameter dan visualisai. Paket `kernlab` untuk mengoptimalkan basis kernel sehingga memberikan hasil implementasi SVM yang fleksibel dan diperluas. Paket `klaR` mengimplementasikan SVM dengan klasifikasi seperti analsis pemisah reguler. Paket terakhir yaitu `svmpath` yang berfungsi untuk menyediakan algoritma yang sesuai dengan hasil solusi pada SVM. *Naïve Bayes* memiliki beberapa paket yang ada pada program R yaitu paket `naïvebayes` dan `e1071`. Paket `naivebayes` merupakan paket yang berfungsi untuk *training* dan *building* pada *Naïve Bayes*. Paket `e1071` merupakan paket *Naïve Bayes* yang berfungsi untuk metode penyelesaian parameter (Karatzouglou dkk., 2006).

# BAB 3. METODOLOGI PENELITIAN

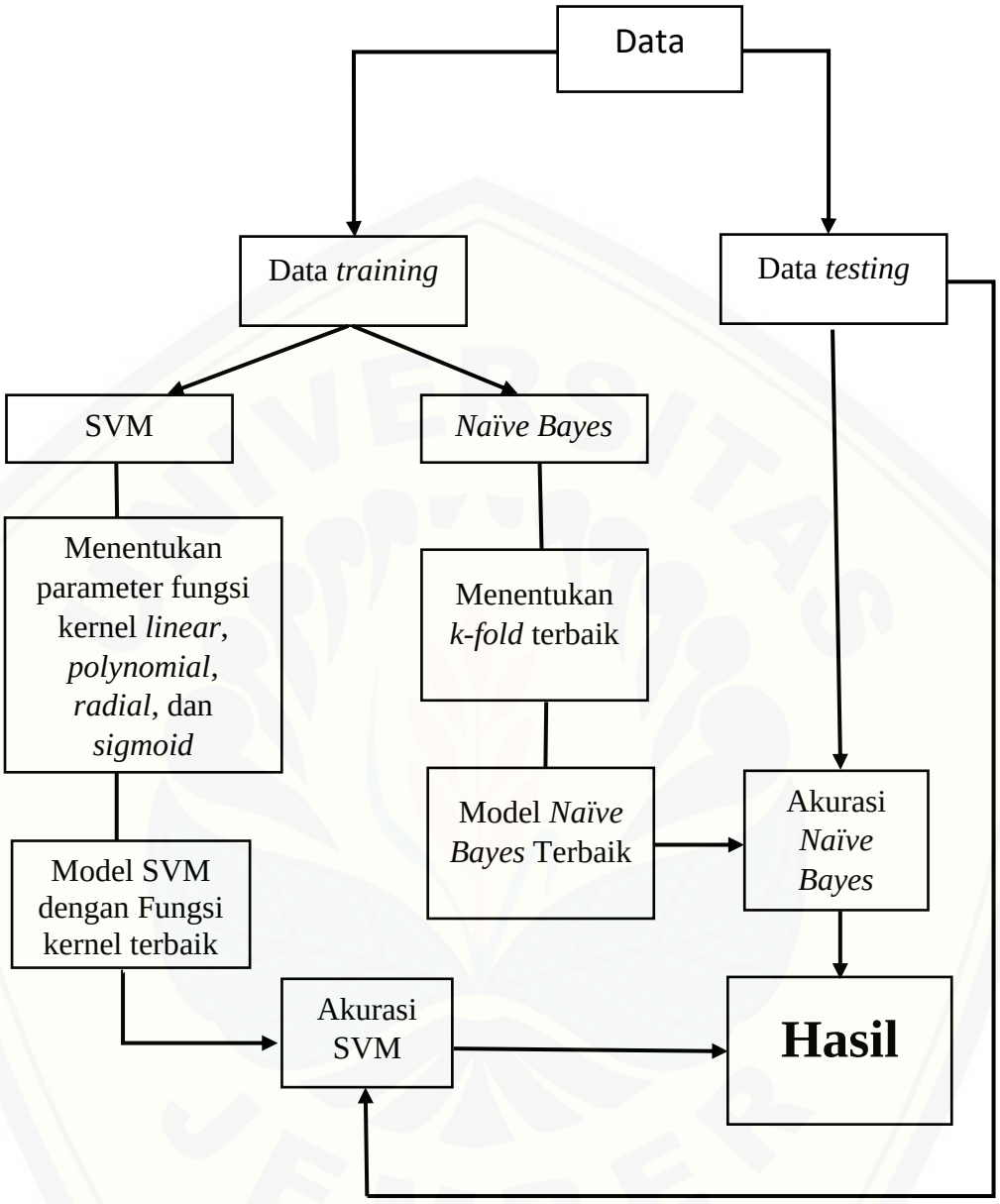
Penelitian ini akan menggunakan data ekspresi gen *microarray*. Jenis data yang digunakan adalah data sekunder yang didapat dari situs https://www.ncbi.nlm.nih.gov. Data yang digunakan sebanyak 80 individu dengan 2408 variabel gen kanker paru yang diklasifikasikan kedalam kelas Normal yang didefinisikan dengan 0 dan kelas Kanker yang didefinisikan dengan 1. Metode analisis data yang digunakan dalam penelitian ini adalah SVM dan *Naïve Bayes* dengan menggunakan *software* R studio. Langkah-langkah yang dilakukan dalam penelitian adalah sebagai berikut:

1. Mengolah data sesuai dengan metode SVM dan *Naive Bayes*.
   Data yang diperoleh disesuaikan menjadi bentuk matriks terlebih dahulu melalui program *excel*. Data yang sesuai dengan metode diinputkan kedalam program R melalui paket `readxl`.
2. Membagi data menjadi dua bagian, *training* dan *testing*.
   Melakukan spliting data 75:25 dengan proporsi sama setiap kelas. Pembagian data menggunakan paket `caret` dengan fungsi `createDataPartition`. Kemudian membuat data frame dari data *training* dan data *testing*.
3. Melakukan Klasifikasi menggunakan *Support Vector Machine* dengan tahapan:
   a. Melakukan uji data *training* pada program R menggunakan paket `e1071` dengan menggunakan fungsi kernel yang akan digunakan, yaitu kernel *Linear, Polynomial, Radial, dan Sigmoid*. Masing-masing fungsi kernel menggunakan *k-fold cross validation* dengan nilai *k* adalah 5 dan 10.
   b. Melakukan *tuning* parameter untuk mencari nilai *cost* terbaik dari hasil uji data *training* pada semua kernel untuk digunakan pada data *testing*.
   c. Melakukan pengujian pada data *testing* menggunakan fungsi kernel yang memiliki nilai *cost* terbaik.

d. Menentukan hasil akurasi dan hasil prediksi klasifikasi kelas normal dan kanker pada eskpresi genetik kanker paru.

4. Melakukan Klasifikasi menggunakan *Naïve Bayes* dengan tahapan:
   a. Melakukan uji data *training* pada program R menggunakan paket `e1071`.
   b. Mencari metode *k-fold cross validation* terbaik dari hasil uji data *training* untuk digunakan pada data *testing*.
   c. Melakukan pengujian pada data *testing* menggunakan nilai *k* terbaik terbaik.
   d. Menentukan hasil akurasi dan hasil prediksi klasifikasi kelas normal dan kanker pada eskpresi genetik kanker paru.
5. Membandingkan hasil klasifikasi model SVM dan model *Naïve Bayes*.

Secara singkat prosedur metode penelitian diuraikan sebagai berikut :




Gambar 3.1 Langkah-langkah penelitian.

# BAB 5. PENUTUP

## 5.1 Kesimpulan

Berdasarkan hasil pengujian metode *Naive bayes* dan SVM terhadap data ekspresi genetika kanker paru menggunakan program R, dapat diambil kesimpulan bahwa :

1. Model pengujian terbaik dari SVM menggunakan kernel *linear* dengan parameter *cost* sebesar 0,001 dan menggunakan metode *5-fold cross validation* dengan ketepatan klasifikasi sebesar 90%.
2. Model pengujian terbaik dari *Naive Bayes* menggunakan metode *10-fold cross validation* dengan ketepatan klasifikasi sebesar 75%.
3. Klasifikasi SVM dari data sel kanker paru menghasilkan 18 data terklasifikasi secara benar dan ada 2 data kesalahan, sedangkan klasifikasi *Naïve Bayes* dari ekspresi genetik sel kanker paru menghasilkan 15 data terklasifikasi secara benar dan ada 5 data kesalahan.

## 5.2 Saran

Pada penelitian ini masih terdapat kekurangan yang dapat diperbaiki pada penelitian selanjutnya. Untuk pengembangan selanjutnya disarankan untuk melakukan penggabungan antara dua metode atau menggunakan metode *Reduced Support Vector Machine* (RSVM) dan *Smooth Support Vector Machine* (SSVM) untuk dataset yang lebih besar.

# DAFTAR PUSTAKA

Ayu, Foriana dan Santi Wulan. 2012. *Analisis Diagnosis Kanker Payudara Menggunakan Regresi Logistik dan Support Vector Machine (SVM) Berdasar Hasil Mamografi*. Surabaya: Institut Teknologi Sepuluh Nopember.

Bramer, Max. 2007. *Principles of Data Mining*. London : Springer.

Cristianini, N dan Taylor J.S. 2000. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge: Cambridge Press University.

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

Han, J dan Kamber, M. 2006. *Data Mining Concept and Tehniques*. San Fransisco : Morgan Kauffman.

Hand, David J. dan YU, Keming. 2001. Idiot's Bayes: Not So Stupid after All?. *International Statistical Review*, 69 (3),hal: 385- 398.

Hidayatul, S. H dan Yuita A. S. 2018. *Seleksi Fitur Information Gain untuk Klasifikasi Penyakit Jantung Menggunakan Kombinasi Metode K-Nearest Neighbor dan Naïve Bayes.* Malang: Universita Brawijaya.

Jusuf, A dkk. 2005. *Kanker paru jenis karsinoma bukan sel kecil. Pedoman Nasional untuk diagnosis dan penatalaksanaan di Indonesia 2005*. Jakarta: PDPI.

Karatzouglou, A. D. Mayer dan K. Hornik. 2006. *Support Vector Machine in R. Journal of Statistic Sofware.* Vol. 15, Issue 9.

Kecman, V. 2005. *Support Vector Machines – an Introduction*. Netherlands: Springer-Verlag Berlin Heidelberg.

Kemenkes RI. 2008. *Laporan Hasil riset kesehatan dasar nasional tahun 2007*. Jakarta : Departemen Kesehatan RI.

Langley dan S. Sage. 1994. *Induction of Selective Bayesian Classifier*. *Proceeding of The Tenth Conference on Uncertainty in Artificial Intelligence.* New York : Morgan Kaufmann.

Lee, Y. dan Mangasarian. O.L. 2001. Support Vector Machine. *Journal of Computational Optimization and Apoooplications*, 20, hal. 5 – 22.

Lewis, D. 1998. Naïve Bayes at forty: *The independence assumption in information retrieval. Proceedings of the Tenth European Conference on Machine Learning*. Berlin, Germany.

Munawarah, R., O. Soesanto dan M. Reza. 2016. *Penerapan Metode Support Vector Machine Pada Diagnosa Hepatitis*. Banjarbaru :Universitas Lampung Mangkurat.

Nugroho, A.S. 2007. *Pengantar Support Vector Machine.* e-tutorial SVM : milis indo-dm@yahoogroups.com.

Nugroho, A.S., Arief B. Witarto dan Dwi Handoko. 2003. *Suport Vector Machines : Teori Aplikasinya dalam Bioinformatika*. Kuliah Umum Ilmu Komputer.com.

Prasetyo, E. 2012. *Data Mining: Konsep dan Aplikasi Menggunakan MATLAB.* Yogyakarta. C.V Andi Offset.

Ridwan, M., Suyono, H., & Sarosa, M. (2013). *Penerapan Data Mining Untuk Evaluasi Kinerja Akademik Mahasiswa Menggunakan Algoritma Naive Bayes Classifier*. Jurnal EECCIS, 59- 64.

Saleh, A. 2015. *Implementasi Metode Klasifikasi Naive Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga.* Journal Citec, 207-217.

World Health Organization. 2018. WHO Media Centre di http://www.who.int/mediacentre/factsheets/fs297/en/. (di akses 20 November 2020).

# LAMPIRAN

## Lampiran A. Data Penelitian Ekspresi Gen Kanker Paru

| No | 1 | 2 | 3 | ... | 2407 | Output |
|---|---|---|---|---|---|---|
| 1 | 134.764.021.798.849 | 158.088.950.971.019 | 83.591.986.270.599 | ... | 674.035.351.293.566 | 1 |
| 2 | 102.995.849.177.268 | 140.517.718.977.555 | 811.352.787.031.939 | ... | 781.918.581.950.247 | 1 |
| 3 | 142.487.876.979.378 | 192.126.829.354.357 | 784.302.933.099.523 | ... | 80.484.875.854.411 | 1 |
| 4 | 116.882.640.866.218 | 220.148.190.169.971 | 708.589.327.032.794 | ... | 722.307.888.811.386 | 1 |
| 5 | 177.611.003.595.112 | 211.349.072.171.003 | 705.866.695.896.195 | ... | 813.166.726.339.286 | 1 |
| 6 | 173.693.603.688.057 | 218.978.886.757.520 | 216.461.293.752.061 | ... | 765.397.811.396.671 | 1 |
| 7 | 107.730.289.220.732 | 140.687.378.117.752 | 74.796.431.585.845 | ... | 686.402.813.542.745 | 1 |
| 8 | 155.028.443.340.059 | 22.412.786.009.528 | 893.436.027.793.957 | ... | 649.809.517.371.073 | 1 |
| 9 | 159.377.482.677.234 | 246.031.874.688.796 | 9.630.012.750.252 | ... | 609.154.963.132.675 | 1 |
| 10 | 150.551.262.414.528 | 199.519.496.572.939 | 932.820.786.043.666 | ... | 81.925.290.099.386 | 1 |
| | | | | ... | | |
| 35 | 255.212.466.380.434 | 367.701.455.523.497 | 960.053.623.778.013 | ... | 131.185.829.444.634 | 1 |
| 36 | 132.307.213.832.810 | 221.576.314.542.387 | 111.432.178.900.641 | ... | 761.492.409.092.575 | 1 |
| 37 | 163.848.205.375.275 | 199.401.281.449.227 | 102.601.191.753.872 | ... | 756.228.555.489.903 | 1 |
| 38 | 179.592.700.571.512 | 234.201.944.698.888 | 937.803.800.713.929 | ... | 663.388.962.519.177 | 1 |
| 39 | 103.478.372.914.747 | 145.104.249.837.849 | 861.188.273.649.914 | ... | 599.127.304.679.347 | 1 |
| 40 | 316.115.656.136.302 | 265.449.139.373.381 | 71.625.209.751.195 | ... | 768.772.508.788.584 | 1 |
| | | | | ... | | |
| 75 | 129.398.189.769.986 | 196.185.953.308.910 | 880.408.312.685.588 | ... | 662.894.271.612.414 | 0 |
| 76 | 100.197.250.852.708 | 162.242.739.844.339 | 71.954.757.051.361 | ... | 653.737.847.723.557 | 0 |
| 77 | 999.032.177.062.888 | 189.986.466.353.922 | 735.347.217.219.104 | ... | 703.228.085.429.637 | 0 |
| 78 | 236.438.636.807.062 | 239.903.764.512.235 | 105.192.052.866.456 | ... | 767.266.847.398.241 | 0 |
| 79 | 101.522.270.741.746 | 146.384.749.066.431 | 742.969.570.680.657 | ... | 789.506.763.141.217 | 0 |
| 80 | 122.167.313.011.322 | 196.155.901.859.047 | 835.254.729.935.012 | ... | 881.499.376.839.476 | 0 |

## Lampiran B. Pemanggilan Data

```
> #Memanggil package yang diperlukan
> library(e1071)
> library(caret)
> library(readxl)
> #Memanggil data
> data=read_excel("G:/XAMPP/Kuliah/Refrensi/DATA/2DATA PENYAKIT P
ARU.xlsx")
> set.seed(123)
```

```
> split = (createDataPartition(y=data$Output, p=0.75, list=FALSE)
)
> training_set = data[split,]
> test_set = data[-split,]
> training_set$Output=as.factor(training_set$Output)
> test_set$Output=as.factor(test_set$Output
```

**Lampiran C1. Pengujian Data Training SVM Kernel Linear**

```
> #Fitting kernel SVM
> tc <- tune.control(cross = 5)
> classifier = svm(formula = Output ~ .,
+                  data = training_set,
+                  type = 'C-classification',
+                  kernel = 'linear', trainControl=tc)
> #Prediction training data result
> y_train_pred = predict(classifier, newdata=training_set[-2408])
> cm_train = table(training_set$Output, y_train_pred)
> success_ratio <- function(cm) {
+   ratio = (sum(cm[1,1]+cm[2,2])/sum(cm))
+   return(ratio)}
> cm_train_str = capture.output(show(cm_train))
> writeLines(c(
+   "Training set confusion matrix : ",
+   cm_train_str,
+   paste("Success ratio on training set : ", toString(success_ra
tio(cm=cm_train)*100), "%")))
Training set confusion matrix :
   y_train_pred
     0  1
  0 18  0
  1  0 42
Success ratio on training set :  100 %
```

**Lampiran C2. Pengujian Data Training SVM Kernel Polynomial**

```
> #Fitting kernel SVM
> tc <- tune.control(cross = 5)
> classifier = svm(formula = Output ~ .,
+                  data = training_set,
+                  type = 'C-classification',
+                  kernel = 'polynomial', trainControl=tc)
> #Prediction training data result
> y_train_pred = predict(classifier, newdata=training_set[-2408])
> cm_train = table(training_set$Output, y_train_pred)
> success_ratio <- function(cm) {
+   ratio = (sum(cm[1,1]+cm[2,2])/sum(cm))
+   return(ratio)}
> cm_train_str = capture.output(show(cm_train))
> writeLines(c(
+   "Training set confusion matrix : ",
+   cm_train_str,
+   paste("Success ratio on training set : ", toString(success_ra
tio(cm=cm_train)*100), "%")))
Training set confusion matrix :
   y_train_pred
     0  1
  0  9  9
  1  0 42
Success ratio on training set :  85 %
```

**Lampiran C3. Pengujian Data Training SVM Kernel Radial**

```
> #Fitting kernel SVM
> tc <- tune.control(cross = 5)
> classifier = svm(formula = Output ~ .,
+                  data = training_set,
+                  type = 'C-classification',
+                  kernel = 'radial', trainControl=tc)
> #Prediction training data result
> y_train_pred = predict(classifier, newdata=training_set[-2408])
> cm_train = table(training_set$Output, y_train_pred)
> success_ratio <- function(cm) {
+   ratio = (sum(cm[1,1]+cm[2,2])/sum(cm))
+   return(ratio)}
> cm_train_str = capture.output(show(cm_train))
> writeLines(c(
+   "Training set confusion matrix : ",
+   cm_train_str,
+   paste("Success ratio on training set : ", toString(success_ra
tio(cm=cm_train)*100), "%")))
Training set confusion matrix :
   y_train_pred
     0  1
  0 18  0
  1  0 42
Success ratio on training set :  100 %
```

**Lampiran C4. Pengujian Data Training SVM Kernel Sigmoid**

```
> #Fitting kernel SVM
> tc <- tune.control(cross = 5)
> classifier = svm(formula = Output ~ .,
+                  data = training_set,
+                  type = 'C-classification',
+                  kernel = 'sigmoid', trainControl=tc)
> #Prediction training data result
> y_train_pred = predict(classifier, newdata=training_set[-2408])
> cm_train = table(training_set$Output, y_train_pred)
> success_ratio <- function(cm) {
+   ratio = (sum(cm[1,1]+cm[2,2])/sum(cm))
+   return(ratio)}
> cm_train_str = capture.output(show(cm_train))
> writeLines(c(
+   "Training set confusion matrix : ",
+   cm_train_str,
+   paste("Success ratio on training set : ", toString(success_ra
tio(cm=cm_train)*100), "%")))
Training set confusion matrix :
   y_train_pred
     0  1
  0 18  0
  1  0 42
Success ratio on training set :  100 %
```

**Lampiran D. Tune parameter SVM**

```
> #mencari cost error
> tuning <- tune(svm,Output~.,data=training_set,kernel="linear",r
anges=list(cost=c(0.001, 0.01, 0.1, 1, 10, 100)),tunecontrol = tc
)
```

```
> tuning

Parameter tuning of ‘svm’:

- sampling method: 5-fold cross validation

- best parameters:
  cost
 0.001

- best performance: 0.1166667

> #mencari cost error
> tuning <- tune(svm,Output~.,data=training_set,kernel="polynomia
l",ranges=list(cost=c(0.001, 0.01, 0.1, 1, 10, 100)),tunecontrol
= tc)
> tuning

Parameter tuning of ‘svm’:

- sampling method: 5-fold cross validation

- best parameters:
  cost
 0.001

- best performance: 0.3

> #mencari cost error
> tuning <- tune(svm,Output~.,data=training_set,kernel="radial",r
anges=list(cost=c(0.001, 0.01, 0.1, 1, 10, 100)),tunecontrol = tc
)
> tuning

Parameter tuning of ‘svm’:

- sampling method: 5-fold cross validation

- best parameters:
  cost
  1

- best performance: 0.266667


> #mencari cost error
> tuning <- tune(svm,Output~.,data=training_set,kernel="sigmoid",
ranges=list(cost=c(0.001, 0.01, 0.1, 1, 10, 100)),tunecontrol = t
c)
> tuning

Parameter tuning of ‘svm’:

- sampling method: 5-fold cross validation

- best parameters:
  cost
  10

- best performance: 0.2
```

```
> #mencari cost error
> tuning <- tune(svm,Output~.,data=training_set,kernel="linear",r
anges=list(cost=c(0.001, 0.01, 0.1, 1, 10, 100)),tunecontrol = tc
)
> tuning

Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
  cost
 0.001

- best performance: 0.1333333


> #mencari cost error
> tuning <- tune(svm,Output~.,data=training_set,kernel="polynomia
l",ranges=list(cost=c(0.001, 0.01, 0.1, 1, 10, 100)),tunecontrol
= tc)
> tuning

Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
  cost
 0.001

- best performance: 0.266666


> #mencari cost error
> tuning <- tune(svm,Output~.,data=training_set,kernel="radial",r
anges=list(cost=c(0.001, 0.01, 0.1, 1, 10, 100)),tunecontrol = tc
)
> tuning

Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
  cost
  10

- best performance: 0.25


> #mencari cost error
> tuning <- tune(svm,Output~.,data=training_set,kernel="sigmoid",
ranges=list(cost=c(0.001, 0.01, 0.1, 1, 10, 100)),tunecontrol = t
c)
> tuning

Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation
```

```
- best parameters:
  cost
  10

- best performance: 0.1333333
```

**Lampiran E. Pengujian data testing SVM**

```
>  #Test SVM_fold
> folds = createFolds(training_set$Output, k = 5)
> cv = lapply(folds, function(x) {
+     training_fold = training_set[-x, ]
+     test_fold = training_set[x, ]
+     classifier = svm(formula = Output ~ .,
+                      data = training_fold,
+                      type = 'C-classification',
+                      kernel = 'linear',cost=0.001)
+     y_pred = predict(classifier, newdata = test_fold[-2408])
+     cm_test= table(test_fold$Output, y_pred)
+     accuracy = (cm_test[1,1] + cm_test[2,2]) / (cm_test[1,1] +
cm_test[2,2] + cm_test[1,2] + cm_test[2,1])
+     return(accuracy)})
> accuracy = mean(as.numeric(cv))
> accuracy
[1] 0.9008159
> cv
$Fold1
[1] 0.9090909

$Fold2
[1] 0.9166667

$Fold3
[1] 0.9090909

$Fold4
[1] 0.9230769

$Fold5
[1] 0.8461538
```

**Lampiran F. Prediksi Klasifikasi SVM**

```
> #Test SVM
> prediksi=predict(classifier,newdata = test_set[-2408])
> cm_test = table(test_set$Output,prediksi)
> cm_test_str = capture.output(show(cm_test))
> writeLines(c(
+ "Test set confusion matrix : ",
+ cm_test_str,
+ paste("Success ratio on training set : ", toString(success_rati
o(cm=cm_test)*100), "%")))
Test set confusion matrix :
   prediksi
     0  1
  0  4  1
  1  1 14
Success ratio on training set :  90 %
```

**Lampiran G. Pengujian data training Naïve Bayes**

```
> tc <- tune.control(cross = 5)
```

```
> classifier = naiveBayes(formula = Output~.,data = training_set,
laplace = 10, trainControl=tc)
> #Prediction training data result
> y_train_pred = predict(classifier, newdata=training_set[-2408])
> cm_train = table(training_set$Output, y_train_pred)
> success_ratio <- function(cm) {
+   ratio = (sum(cm[1,1]+cm[2,2])/sum(cm))
+   return(ratio)}
> cm_train_str = capture.output(show(cm_train))
> writeLines(c(
+   "Training set confusion matrix : ",
+   cm_train_str,
+   paste("Success ratio on training set : ", toString(success_ra
tio(cm=cm_train)*100), "%")))
Training set confusion matrix :
   y_train_pred
     0  1
  0 17  1
  1  0 42
Success ratio on training set :  98.3333333333333 %
```

**Lampiran I. Pengujian Data testing Naïve Bayes**

```
#Test NB_fold
> folds = createFolds(training_set$Output, k = 5)
> cv = lapply(folds, function(x) {
+   training_fold = training_set[-x, ]
+   test_fold = training_set[x, ]
+   classifier = naiveBayes(formula = Output ~ .,
+                           data = training_fold,
+                           laplace = 10,threshold=0.01,
eps=0.001)
+   y_pred = predict(classifier, newdata = test_fold[-2408])
+   cm_test= table(test_fold$Output, y_pred)
+   accuracy = (cm_test[1,1] + cm_test[2,2]) / (cm_test[1,1] +
cm_test[2,2] + cm_test[1,2] + cm_test[2,1])
+   return(accuracy)})
> accuracy = mean(as.numeric(cv))
> accuracy
[1] 0.7700466
> cv
$Fold1
[1] 0.6153846

$Fold2
[1] 0.75

$Fold3
[1] 0.8333333

$Fold4
[1] 0.8333333

$Fold5
[1] 0.8181818

> #Test NB_fold
> folds = createFolds(training_set$Output, k = 10)
> cv = lapply(folds, function(x) {
+   training_fold = training_set[-x, ]
+   test_fold = training_set[x, ]
+   classifier = naiveBayes(formula = Output ~ .,
+                           data = training_fold,
```

```
+                             laplace = 10,threshold=0.01, eps=0.00
1)
+   y_pred = predict(classifier, newdata = test_fold[-2408])
+   cm_test= table(test_fold$Output, y_pred)
+   accuracy = (cm_test[1,1] + cm_test[2,2]) / (cm_test[1,1] + cm
_test[2,2] + cm_test[1,2] + cm_test[2,1])
+   return(accuracy)})
> accuracy = mean(as.numeric(cv))
> accuracy
[1] 0.8114286
> cv
$Fold01
[1] 1

$Fold02
[1] 1

$Fold03
[1] 0.6

$Fold04
[1] 0.6666667

$Fold05
[1] 0.8571429

$Fold06
[1] 0.8

$Fold07
[1] 0.8333333

$Fold08
[1] 0.6666667

$Fold09
[1] 0.8571429

$Fold10
[1] 0.8333333
```

**Lampiran J. Prediksi Klasifikasi Naïve Bayes**

```
> #Test NB
> prediksi=predict(classifier,newdata = test_set[-2408])
> cm_test = table(test_set$Output,prediksi)
> cm_test_str = capture.output(show(cm_test))
> writeLines(c(
+   "Test set confusion matrix : ",
+   cm_test_str,
+   paste("Success ratio on training set : ", toString(success_ra
tio(cm=cm_test)*100), "%")))
Test set confusion matrix :
   prediksi
     0  1
  0  3  2
  1  3 12
Success ratio on training set :  75 %
```