# KLASIFIKASI HASIL PERTANDINGAN TIM SEPAK BOLA MENGGUNAKAN METODE *RANDOM FOREST* DAN *SUPPORT VECTOR MACHINE*

(Studi Kasus: Data Pertandingan Liga 1 Indonesia Musim 2018)

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Program Studi Statistika

**Alfari Afdhal**

**15 611 044**

**JURUSAN STATISTIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS ISLAM INDONESIA**

**YOGYAKARTA**

**2020**

# HALAMAN PERSETUJUAN PEMBIMBING

## TUGAS AKHIR

| | | |
|---|---|---|
| Judul | : | Klasifikasi Hasil Pertandingan Tim Sepak Bola Menggunakan Metode *Random Forest* Dan *Support Vector Machine* (Studi Kasus: Data Pertandingan Liga 1 Indonesia Musim 2018) |
| Nama Mahasiswa | : | Alfari Afdhal |
| Nomor Mahasiswa | : | 15611044 |

TUGAS AKHIR INI TELAH DIPERIKSA DAN DISETUJUI UNTUK DIUJIKAN

Yogyakarta, 23 September 2020

Pembimbing

**Muhammad Muhajir, S.Si., M.Sc.**

# HALAMAN PENGESAHAN

## TUGAS AKHIR

## KLASIFIKASI HASIL PERTANDINGAN TIM SEPAK BOLA MENGGUNAKAN METODE *RANDOM FOREST* DAN *SUPPORT VECTOR MACHINE*

(Studi Kasus: Data Pertandingan Liga 1 Indonesia Musim 2018)

Nama Mahasiswa : Alfari Afdhal

Nomor Mahasiswa : 15611044

**TUGAS AKHIR INI TELAH DIUJIKAN**

**PADA TANGGAL 5 OKTOBER 2020**

| **Nama Penguji** | **Tanda Tangan** |
|---|---|
| 1. Muhammad Hasan Sidiq K, S.Si., M.Sc. | ..... |
| 2. Rahmadi Yotenka, S.Si., M.Sc. | ..... |
| 3. Muhammad Muhajir, S.Si., M.Sc. | ..... |

Mengetahui,

Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam

**Prof. Riyanto, S.Pd., M.Si., Ph.D.**

# KATA PENGANTAR

***Assalamualaikum Wr.Wb.***

*Alhamdulillahirobbil alamin*, segala puji dan syukur kehadirat Allah SWT yang telah mencurahkan rahmat, nikmat dan hidayah-Nya selama penyusunan Tugas Akhir ini sehingga dapat diselesaikan. Tidak lupa pula shalawat serta salam tercurah kepada Nabi Muhammad SAW beserta keluarga, sahabat dan para pengikut-pengikutnya, karena berkat usaha beliau kita dapat terbebas dari zaman jahiliyah menuju zaman ilmu pengetahuan seperti yang kita rasakan pada saat ini.

Tugas Akhir yang berjudul "**Klasifikasi Hasil Pertandingan Tim Sepak Bola Menggunakan Metode *Random Forest* dan *Support Vector Machine* (Studi Kasus: Data Pertandingan Liga 1 Indonesia Musim 2018)**" ini merupakan salah satu syarat untuk memperoleh gelar sarjana Program Studi Statistika di Universitas Islam Indonesia. Dalam penyusunan Tugas Akhir ini penulis melewati berbagai macam hambatan dan rintangan namun penulis beruntung telah banyak mendapat bimbingan dan bantuan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis bermaksud menyampaikan ucapan terima kasih kepada :

1. Bapak Prof. Riyanto, S.Pd., M.Si., Ph.D. selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Islam Indonesia, Yogyakarta beserta seluruh jajarannya.
2. Bapak Dr. Edy Widodo, S.Si., M.Si., selaku Ketua Program Studi Statistika beserta seluruh jajarannya.
3. Bapak Muhammad Muhajir, S.Si., M.Sc., selaku Dosen Pembimbing penulis yang sangat berjasa dan sangat sabar dalam membimbing penulis dalam menyelesaikan Tugas Akhir ini.
4. Dosen-dosen Statistika Universitas Islam Indonesia yang telah mengajar, memberikan ilmunya serta motivasi dan inspirasi bagi penulis.

5. Kedua orang tua tersayang dan tercinta, Bapak H. Amri dan Ibu Hj. Fitra Teti yang tiada hentinya selalu memberikan doa, dukungan, motivasi dan semangat dalam menyelesaikan Tugas Akhir ini.
6. Sahabat satu kontrakan yaitu Yulan, Bayu, Anugrah, Panji dan Ihsan teman seperjuangan selama 4 tahun di Yogyakarta.
7. Teman-teman Statistika UII Angkatan 15 "*Affinistic*" yang telah banyak membantu penulis diberbagai kesempatan baik akademik maupun non akademik.
8. Keluarga besar IKS yang telah banyak membantu dalam perkuliahan maupun non perkuliahan serta sering memberikan pelayanan *bank* soal ujian UTS dan UAS.
9. Teman-teman KKN UII 57 Desa Pelutan yang telah memberikan banyak kenangan selama masa KKN serta merasakan pengalaman di masyarakat secara langsung.
10. Ibu Kos Bu Menuk yang selalu ramah terhadap penulis dan sering memberikan makanan dikala penulis kelaparan.
11. Semua pihak yang tidak dapat disebutkan satu per satu, baik yang memberikan dukungan dan bantuan secara langsung maupun tidak langsung yang penulis sadari atau tidak maka untuk itu semua penulis mengucapkan terima kasih sebanyak-banyaknya.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari sempurna, oleh karena itu segala kritik dan saran yang bersifat membangun selalu penulis harapkan. Semoga Tugas Akhir ini dapat bermanfaat bagi penulis khususnya dan bagi semua yang membutuhkan. Akhir kata, semoga Allah SWT selalu melimpahkan rahmat serta hidayah-Nya kepada kita semua, Aamiin aamiin ya robbal'alamin.

***Wassalamu'alaikum, Wr.Wb***

Yogyakarta, 6 September 2020

(Alfari Afdhal)

# DAFTAR ISI

# DAFTAR TABEL

# DAFTAR GAMBAR

# DAFTAR LAMPIRAN

# PERNYATAAN

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang sebelumnya pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali diacu dalam naskah ini dan diterbitkan dalam daftar pustaka.

Yogyakarta, 6 September 2020


Alfari Afdhal

# KLASIFIKASI HASIL PERTANDINGAN TIM SEPAK BOLA MENGGUNAKAN METODE *RANDOM FOREST* DAN *SUPPORT VECTOR MACHINE*

(Studi Kasus: Data Pertandingan Liga 1 Indonesia Musim 2018)

Oleh: Alfari Afdhal
Program Studi Statistika Fakultas MIPA
Universitas Islam Indonesia

## INTISARI

Dalam dunia sepakbola, terdapat banyak faktor-faktor yang mempengaruhi kemenangan suatu tim dalam memenangkan pertandingan, oleh karenanya disini diperlukan klasifikasi terhadap hal tersebut agar memudahkan untuk melihat mana faktor yang paling berpengaruh terhadap kemenangan tim. Salah satu cara yang bisa digunakan adalah metode *random forest* dan juga *support vector machine*. Metode ini dapat mengurutkan mana variabel yang paling berpengaruh dibandingkan dengan variabel yang lain berdasarkan dari data yang digunakan, serta menentukan seberapa besar nilai variabel tersebut untuk membuat peluang kemenangan tim menjadi lebih besar. Dari hasil analisis yang dilakukan dengan data yang ada, didapatkanlah bahwa beberapa faktor yang mempengaruhi tim kemenangan suatu tim dalam memenangkan pertandingan adalah jumlah tembakan *on target* lalu diikuti oleh akurasi operan, tekel sukses, sepak pojok, pelanggaran, total tembakan, *offside*, kartu kuning, dan kartu merah yang kesetiap variabel tersebut mempunyai besar pengaruh yang berbeda-beda.

**Kata Kunci :** Sepakbola, Klasifikasi, *Random Forest*, *Support Vector Machine.*

***CLASSIFICATION OF THE RESULT OF THE FOOTBALL TEAM USING RANDOM FOREST AND SUPPORT VECTOR MACHINE METHOD***

*(Case Study: Data Match* Liga 1 Indonesia *Season* 2018*)*

*By:* Alfari Afdhal

*Departement of Statistics, Faculty of Mathematics and Natural Sciences*

*Islamic University of Indonesia*

## *ABSTRACT*

*In the world of football, there are many factors that influence the victory of a team in winning a match, therefore a classification is needed here to make it easier to see which factors have the most influence on team victory. One method that can be used is the random forest method and also support vector machines. This method can sort which variable is the most influential compared to other variables based on the data used, and determine how much the value of these variables is to make the team's chances of winning greater. From the results of the analysis carried out with existing data, it was found that several factors that influenced the winning team of a team in winning the match were the number of shots on target followed by passing accuracy, successful tackles, corners, violations, total shots, offside, yellow cards, and the red card which each variable has a different effect.*

***Keywords:*** *Football, Classification, Random Forest, Support Vector Machine.*

# BAB I
# PENDAHULUAN

## 1.1 Latar Belakang

Statistika merupakan pengetahuan yang berhubungan dengan cara pengumpulan data, penyusunan data, pengolahan data, penganalisisan data, serta penyajian data berdasarkan kumpulan dan analisis data yang dilakukan. Di sepakbola, statistik digunakan untuk menganalisis kemampuan pemain, perkiraan strategi suatu tim, analisis pertandingan, *scouting* pemain serta kebutuhan data tim dan masih banyak lagi. Di dunia sepakbolaan modern penggunaan statistik sudah marak dilakukan, bahkan sudah menjadi konsumsi publik sehari-hari. Statistik juga merupakan bentuk sederhana dari ilmu sains yang dikombinasikan dengan ilmu olahraga, bisa dikatakan juga *sports science* (Putra Eka, 2017).

Manfaat statistik yang paling besar adalah bagi pelatih. Selain berguna untuk meningkatkan kompetensi tim, statistika juga digunakan oleh para pelatih untuk melihat kekuatan calon lawan. Dengan begitu diharapkan tim yang diasuhnya mampu meraih kemenangan. Bukan hanya pada saat sebelum pertandingan saja, tetapi juga pada saat pertandingan. Pelatih memperhatikan bagaimana jalannya pertandingan yang sedang berlangsung, lalu memberikan instruksi kepada pemainnya apa yang harus dilakukan, tentunya dengan membaca statistik pertandingan tersebut. Dan pada saat pertandingan usai, statistika menjadi sarana yang tepat untuk mengadakan evaluasi tim, agar tim itu menjadi lebih baik pada pertandingan-pertandingan selanjutnya (Angga, 2013).

Sepak bola Indonesia dalam hal ini klub-klub yang bermain di Liga 1 Indonesia, sebagaian besar masih jarang menggunakan analisis data dalam pengambilan keputusan klub, hal ini dapat dimengerti karena kualitas kompetisi di Indonesia masih belum setara dengan sepak bola eropa yang menjadi kiblat sepak bola dunia, bahkan pada tingkat asia dikutip dari situs resmi asosiasi sepak bola asia

AFC, liga Indonesia masih peringkat 28 asia dan peringkat 7 asia tenggara. (AFC, 2020).

Dalam suatu tim membeli pemain tidak menjadi jaminan suatu klub akan memenangkan banyak gelar, sebab banyak faktor lain yang terlibat pada suatu tim dalam memenangkan pertadingan. Hal inilah yang perlu diamati dengan seksama apa yang menjadi faktor yang paling penting bagi setiap klub untuk memenangkan pertandingan. Terdapat banyak sekali kasus diluar dugaan yang terjadi pada sepakbola, contohnya adalah Yunani yang mampu memenangkan Euro 2004 yang mengalahkan Portugal di final. Dalam konteks prestasi, Leicester City menjadi bukti sahih ampuhnya peran analisis data di sepak bola. Sebelumnya, banyak orang yang menganggap kesuksesan Leicester menjuarai Liga Inggris pada 2016 adalah karena keberuntungan. Namun, di balik prestasi fenomenal itu, ada statistika dan mesin rumit yang bekerja. Leicester merupakan salah satu tim di Inggris yang paling getol memakai jasa *Prozone Sports*, perusahaan teknologi dan analisis data olahraga asal Inggris (Harjono, 2018).

Jika berbicara mengenai faktor-faktor yang berpengaruh dalam suatu pertandingan sepak bola, maka disini diperlukan klasifikasi agar memudahkan untuk melihat mana faktor yang paling berpengaruh terhadap kemenangan tim dan salah satu cara yang bisa digunakan adalah metode *random forest* dan juga *support vector machine*, metode ini dapat mengurutkan mana variabel yang paling berpengaruh dibandingkan dengan variabel yang lain yang tersedia pada data yang digunakan, serta menentukan seberapa besar nilai variabel tersebut untuk membuat peluang kemenangan tim lebih besar dibandingkan peluang tim tidak memenangkan pertandingan.

Banyak sekali jenis metode yang bisa digunakan pada klasifikasi tapi disini peneliti menggunakan *random forest* karena metode ini sangat baik dan mampu mengolah data dalam jumlah besar serta data yang memiliki jumlah variabel yang banyak. Disisi lain metode ini juga tidak terikat dengan distribusi normal pada datanya sehingga hal tersebut tidak menjadi penghalang dalam melakukan analisis. Selanjutnya peneliti juga menggunakan metode *support vector machine* sebagai metode pembanding terhadapa metode *random forest*. Hal ini dipilih dikarenakan

metode *support vector machine* secara umum tergolong metode yang mudah dan simpel digunakan.

Variabel penelitian yang digunakan pada penelitian ini yaitu terdiri dari total tembakan, tembakan *on target*, akurasi operan, tekel sukses, sepak pojok, pelanggaran, *offside*, kartu kuning, dan kartu merah. Kesemua variabel diatas digunakan berdasarkan dari referensi-referensi yang ada seperti dari penelitian Putra (2017) serta Laia, dkk (2019). Selain dengan hal itu disini peneliti juga menyesuaikan dengan data-data yang tersedia di lapangan untuk proses analisisnya.

Adapun keunggulan pada penelitian kali ini adalah studi kasus yang diangkat adalah Liga Indonesia, yang mana penelitian tentang hal ini masih sangat jarang ditemui sehingga bisa menjadi acuan bagi peneliti lain atau bahkan pelatih-pelatih sepakbola di Indonesia untuk melihat keadaan Liga Indonesia jika ditinjau dari segi data statistiknya.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang, permasalahan yang dapat diidentifikasi penulis dalam penelitian ini adalah:

1. Faktor-faktor apa saja yang mempengaruhi kemenangan suatu tim dalam pertandingan sepak bola?
2. Mana metode yang paling tepat untuk dilakukan klasifikasi dari data yang digunakan?
3. Dan berdasarkan faktor yang ada, manakah variabel yang paling berpengaruh terhadap kemenangan tim sepak bola dalam memenangkan pertandingan?

## 1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini tidak terlalu meluas, maka dalam penelitian ini diberikan batasan-batasan sebagai berikut :

1. Data yang digunakan yaitu data hasil pertandingan sepak bola Liga 1 Indonesia musim kompetisi 2018.
2. Metode yang digunakan dalam penelitian ini adalah *random forest* dan *support vector machine*.
3. Data dianalisis menggunakan *Microsoft Excel* 2019 dan *R Studio* 1.2.1335

## 1.4 Tujuan Penelitian

Tujuan penelitian yang ingin dicapai pada penelitian ini adalah :

1. Untuk mengetahui faktor-faktor apa saja yang mempengaruhi kemenangan suatu tim dalam pertandingan sepak bola.
2. Untuk mengetahui seberapa besar faktor tersebut berpengaruh terhadap peluang tim dalam memenangkan pertandingan.
3. Untuk mengetahui faktor manakah yang paling berpengaruh terhadap kemenangan tim sepak bola serta peluang tim dalam memenangkan pertandingan.

## 1.5 Manfaat Penelitian

Adapun manfaat penelitian ini sebagai berikut :

1. Menambah pengetahuan tentang pengklasifikasian variabel yang ada pada olahraga sepak bola dengan menggunakan metode *random forest* dan *support vector machine*.
2. Memberikan pengetahuan tentang besar kecilnya pengaruh suatu variabel terhadap hasil pertandingan sepak bola khususnya di Liga Indonesia.
3. Memberikan informasi tambahan bagi klub dalam merancang strategi permainan dalam menghadapi strategi permainan lawan sehingga efektif dalam setiap pertandingan.

# BAB II
# TINJAUAN PUSTAKA

## 1.6 Penelitian Terdahulu

Terdapat beberapa tinjauan pustaka dari penelitian sebelumnya yang penulis gunakan dalam penelitian ini. Putra (2017) melakukan penelitian menggunakan pohon klasifikasi. Variabel penelitian yang digunakan yaitu data hasil pertandingan liga spanyol yang mana variabel dependennya adalah hasil pertandingan tuan rumah dan untuk variabel independennya adalah selisih dari rata-rata umur pemain, jumlah pemain, total *market value*, jumlah pemain asing, jumlah pemain lokal, transfer pemain masuk, transfer pemain keluar, pengeluaran belanja pemain, peringkat diminggu sebelumnya, umur pelatih, gol dipertandingan sebelumnya dan kebobolan dipertandingan sebelumnya. Berdasarkan hasil penelitian yang dilakukan Putra (2017) diperoleh bahwa terdapat dua pohon klasifikasi yang dihasilkan, pada pohon klasifikasi pertama didapatkan bahwa variabel selisih dari nilai *market value* memiliki pengaruh yang paling besar terhadap kemenangan tim dengan memiliki nilai kesesuai sebesar 56,1%, sedangkan pada pohon klasifikasi kedua didapatkan bahwa variabel selisih dari pengeluaran belanja kedua tim memiliki pengaruh yang paling besar terhadap kemenangan tim dengan memiliki nilai kesesuai sebesar 54,7%.

Laia, dkk (2019) melakukan penelitian menggunakan algoritma C.45. Penelitian tersebut menggunakan variabel dependennya yaitu tim yang akan memenangkan liga *champion*, sedangkan variabel independen yang digunakan yaitu kualitas pemain, umur, kualitas pelatih, finansial, dan banyanya prestasi klub. Berdasarkan penelitian yang dilakukan oleh Laia, dkk (2019) diperoleh variabel yang paling berpengaruh terhadap klub yang akan memenangkan gelar liga *champion* adalah kondisi manajemen finansial klub dan umur rata-rata para pemain sepak bola dalam klub tersebut.

Dewi (2011) melakukan penelitian menggunakan metode *random forest* dalam *driver analysis*. Penelitian tersebut menggunakan data suatu perusahaan riset di Indonesia yang terdiri atas sejumlah merek yang berbeda dari jenis suatu produk yang sama yaitu produk Z. Banyaknya jumlah data yaitu 1200 yang terdiri dari satu variabel dependen yaitu berupa status kesedian orang untuk membeli produk Z dan variabel independen yaitu berupa status kesetujuan seseorang terhadap atribut produk Z. Dari hasil penelitian tersebut didapatkan akurasi prediksi yang tinggi dan stabil saat *random forest* dibangun menggunakan jumlah peubah penjelas sebesar 4 dan jumlah pohon yang digunakan sebesar 500. Pada kondisi tersebut didapat tingkat misklasifikasi yang dicapai sekitar 33%-35,5% dengan nilai rataan sebesar 34,5%.

Octaviani, dkk (2014) melakukan penelitian menggunakan metode klasifikasi *support vector machine*. Penelitian tersebut menggunakan data tentang nilai akreditasi Sekolah Dasar (SD) di Kabupatan Magelang mulai tahun penetapan 2011-2013 yang diperoleh dari *website* resmi BAN-S/M dengan status akreditasi A, B, dan C yang mana ketiga kategori ini dijadikan sebagai variabel dependen dan untuk variabel independennya terdiri dari 7 komponen yaitu standar isi, proses, kompetensi lulusan, pendidik dan tenaga kependidikan, sarana dan prasarana, pengelolaan, dan penilaian Pendidikan. Dari hasil penelitian tersebut dengan menggunakan data *training* sebanyak 337 data didapatkan tingkat akurasi sebesar 100% menggunakan fungsi *kernel Gaussian Radial Basic Function* (RBF) sedangkan menggunakan fungsi *kernel polynomial* akurasi klasifikasi adalah sebesar 92,68%. Pada pengujian dengan data *testing* sebanyak 82 data, akurasi klasifikasi yang didapat yaitu sebesar 93,902% menggunakan fungsi *kernel Gaussian Radial Basic Function* (RBF) sedangkan dengan menggunakan fungsi *kernel polynomial* akurasi klasifikasi adalah sebesar 92,683%.

## 1.7 Perbandingan dengan Penelitian Terdahulu

Berikut merupakan perbandingan secara umum antara penelitian ini dengan penelitian-penelitian yang dilakukan sebelumnya yang ditampilkan pada Tabel 2.1

**Tabel 2. 1** Perbandingan dengan Penelitian Terdahulu

| No. | Nama Peneliti | Judul Penelitian | Persamaan | Perbedaan |
|---|---|---|---|---|
| 1. | Erzha Aulia Putra | Evaluasi Faktor-Faktor Yang Mempengaruhi Kemenangan Dalam Pertandingan Sepak Bola Dengan Menggunakan Pohon Klasifikasi | Topik yang diangkat sama-sama mengenai pengklasifikasian variabel-variabel yang berpengaruh terhadap kemenangan tim sepak bola. | Liga yang diteliti adalah liga Spanyol serta metode yang digunakan adalah pohon klasifikasi. |
| 2. | Yonata Laia, Charles Tandian, dan Andi Saputra | Penerapan Data Mining Dalam Memprediksi Pemenang Klub Sepak Bola Pada Ajang Liga *Champion* Dengan Algoritma C.45 | Topik yang diangkat sama-sama mengenai penentuan variabel-variabel yang berpengaruh terhadap kemenangan tim sepak bola. | Liga yang diteliti adalah liga *Champion* serta metode yang digunakan adalah algoritma C.45. |
| 3. | Nariswari Karina Dewi | Penerapan Metode *Random Forest* Dalam *Driver Analysis* | Metode yang digunakan sama-sama menggunakan metode *random forest* untuk mencari variabel yang paling berpengatuh. | Data yang digunakan pada penelitian ini menggunakan data suatu perusahaan riset di Indonesia. |
| 4. | Pusphita Anna Octaviani, Yuciana Wilandari, dan Dwi Ispriyanti | Penerapan Metode Klasifikasi *Support Vector Machine* (SVM) Pada Data Akreditasi Sekolah Dasar (SD) Di Kabupaten Magelang | Metode yang digunakan sama-sama menggunakan metode *Support Vector Machine*. | Penelitian ini menggunakan data tentang nilai akreditasi Sekolah Dasar (SD) di Kabupatan Magelang mulai tahun penetapan 2011-2013. |

# BAB III
# LANDASAN TEORI

## 2.1 Statistika dalam Sepak Bola

Penggunaan pendekatan ilmiah pada olahraga sepak bola bukanlah hal yang baru dilakukan seperti pada masa sekarang, pada pertengahan tahun 1970an terdapat seorang pelatih legendaris Ukraina bernama Valeriy Lobanovskyi (1939-2002) yang menjadi salah satu pelopor penggunaan ilmu statistika ke dalam dunia sepak bola. Dengan peneliti dan ilmuwan Uni Soviet pada masa itu, beliau merumuskan bagaimana strategi terbaik, pemilihan pemain untuk memperkuat performa tim, serta program pelatihan yang tepat. Berkat jasa-jasanya, Lobanovcskyi dinobatkan sebagai salah satu pelatih terbaik dan mendapat penghargaan FIFA *Order of Merit* yang merupakan penghargaan tertinggi yang diberikan oleh FIFA (badan sepak bola dunia) karena telah memberikan pengaruh dan manfaat yang besar dalam dunia sepak bola (Angga, 2013).

Kehadiran statistika dalam sepak bola mempunyai peran penting dalam berbagai aspek yang ada. Pemain, pelatih, klub, sponsor, hingga para penonton sepak bola sekalipun dapat mengambil manfaat dari adanya data statistik sepak bola. Dari berbagai hal tersebut tentunya yang paling merasakan manfaatnya adalah para pelatih sepak bola, bagi pelatih data-data statistik berguna untuk meningkatkan kekuatan tim serta menganalisa taktik dan kemampuan calon lawan. Disamping itu pelatih juga mampu melihat kekurangan tim dan merekrut pemain baru yang sesuai dengan kebutuhan tim dan di era modern seperti sekarang metode ini sering dilakukan seperti misalnya mantan pelatih Arsenal, Arsene Wenger yang mana pada masa kejayaannya menggunakan data statistik untuk menentukan pemain mana yang akan dibeli berdasarkan karakteristik yang diinginkan dan bukan tanpa sebab beliau dijuluki "*The Professor*". Wenger mempelajari dengan baik statistik permainan pemain muda milik AC Milan dikala itu yang bernama Patrick Vieira dan saat dia disia-siakan di Italia, tanda ragu Wenger membawanya ke London untuk bergabung bersama Arsenal. Sama halnya dengan Thierry Henry yang

terpaku di Juventus, Henry ragu apakah dia cocok untuk dijadikan striker tapi analisa Wenger menunjukkan bahwa dia lebih cocok untuk dijadikan ujung tombak dibandingkan sebagai pemain sayap (Angga, 2013).

Pada kondisi sepak bola di Indonesia, penggunaan ilmu statistika pernah diterapkan pada Timnas U-19 yang bermain pada kejuaraan AFF U-19 tahun 2013 era Evan Dimas dkk. Badan Tim Nasional (BTN) menyadari bahwa statistik permainan sangat diperlukan oleh Timnas Indonesia sehingga dibentuklah *High Performance Unity* (HPU) dan Lab Bola yang mempunyai tanggung jawab mencatat data-data pemain Timnas untuk dijadikan dasar pengambilan keputusan, pelatih Timnas U-19 saat itu yaitu Indra Sjafri mengambil keputusan berdasarkan statistik para pemain yang mana terbukti mampu menciptakan permainan yang sangat baik dan menghibur dan pada akhirnya mampu membawa Timnas U-19 Indonesia memenangkan kejuaraan AFF U-19 tersebut untuk pertama kalinya dalam sejarah sepak bola Indonesia (Yasinaron, 2017).

## 2.2 Analisis Statistika Deskriptif

Menurut Walpole dan Myers (1995) metode statistik merupakan langkah-langkah yang dipakai dalam pengumpulan, penyajian, analisis, dan penafsiran data. Kemudian metode tersebut dibagi dua yaitu statistika deskriptif dan statistika inferensial. Statistika deskriptif merupakan bagian dari ilmu statistik yang meringkas, menyajikan, dan mendeskripsikan data dalam bentuk yang mudah dipahami oleh orang awam tanpa mengurangi informasi yang ada. Statistika deskriptif hanya memberikan informasi mengenai suatu data yang dimiliki tanpa menarik suatu kesimpulan dari hasil data tersebut. Umumnya terdapat dua metode yang biasa dipakai untuk menjelaskan karakteristik suatu data, yaitu:

1. Tabel

   Pada penyajian data dalam bentuk tabel biasanya bertujuan untuk mengelompokkan nilai-nilai pengamatan ke dalam beberapa kelompok yang masing-masing memiliki karakteristik yang sama. Tabel distribusi frekuensi, tabel distribusi frekuensi relatif dan tabel kontingensi adalah bentuk tabel yang

sering digunakan untuk data kualitatif dengan banyak kategori dalam baris maupun kolom.

2. Grafik atau diagram

   Pada penyajian data dalam bentuk grafik atau diagram bertujuan untuk menggambarkan data secara keseluruhan dengan menampilkan karakteristik tertentu dari data yang dimiliki. Histogram, diagram batang dan daun, diagram baris, diagram lingkaran, dan diagram kotak adalah beberapa bentuk jenis grafik atau diagram yang sering digunakan.

## 2.3 *Data Mining*

Menurut Gorunescu (2011) *data mining* merupakan proses mengolah atau merangkum data yang memiliki jumlah yang besar melalui suatu proses analisis agar didapatkan data yang memiliki informasi yang berharga. Selain itu bisa juga diartikan dengan gabungan antara metode statistik dan *artificial intelligence*/kecerdasan buatan yang dapat terus berkembang. Secara sederhana *data mining* adalah penambangan informasi baru dengan mencari suatu bentuk pola dan aturan tertentu dari sekumpulan data dalam jumlah yang sangat besar.

### 2.3.1 Model *Supervised Learning*

Model ini digunakan untuk memprediksi kemungkinan di masa depan berdasarkan data-data yang dimiliki sebelumnya yang dipelajari menggunakan suatu metode tertentu sehingga didapatkan hasil yang akurat. Contoh penggunaan metode *supervised learning* ini misalnya untuk memprediksi kemungkinan terjadinya bahaya bencana alam pada suatu daerah dengan melihat beberapa faktor berdasarkan pada data-data historis sebelumnya yang sudah dipelajari seperti misalnya banjir, gunung meletus, dan lainnya (Jain, 2015).

Menurut Chandra (2017) *supervised learning* adalah sebuah pendekatan dengan cara melatih suatu data dengan menggunakan variabel yang ada kemudian dilakukan pengelompokan data berdasarkan data yang sudah ada. Adapun contoh metode yang menggunakan *supervised learning* antara lain yaitu *Artificial Neural Network, Support Vector Machine, Decission Tree, Nearest Neighbor Classifier, Random Forest, Fuzzy K-Nearest Neighbor,* dan *Naive Bayes Classifier*.

### 2.3.2 Model *Unsupervised Learning*

Menurut Chandra (2017) *unsupervised learning* tidak mempunyai data latih sehingga berdasarkan dari data yang ada dilakukan pengelompokan data menjadi dua bagian atau tiga bagian dan seterusnya. Adapun contoh sederhana dari penggunaan metode yang menggunakan *unsupervised learning* adalah seseorang yang belum pernah membeli buku dan suatu ketika dia membeli buku dalam jumlah yang banyak dengan berbagai macam jenis, agar mudah membedakan dan nantinya mudah dicari maka semua buku itu dilakukan pengelompokan kedalam beberapa kategori dengan mengidentifikasi buku mana yang memiliki kemiripan isi agar dikelompokkan kedalam kategori yang sama. Metode analisis *unsupervised learning* adalah *DBSCAN, Hierarchical Clustering, K-Means, Self-Organizing Map*, dan *Fuzzy C-Means*.

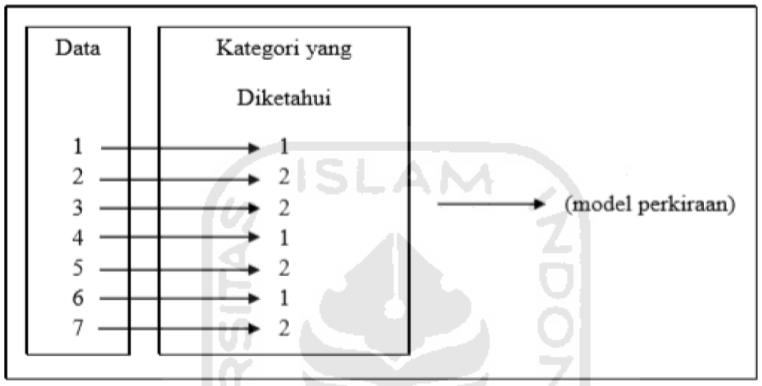
## 2.4 *Machine Learning*

Menurut Santoso (2007) *machine learning* pada dasarnya adalah proses komputer dalam mempelajari data-data yang ada dengan melakukan interaksi yang dilakukan secara berulang-ulang sehingga pola data dapat dikenali. *Machine learning* dibekali kecerdasan buatan yang dibuat oleh *user* dan dimasukan ke dalam komputer. *Machine learning* adalah disiplin ilmu yang menggabungkan pembuatan dan pengembangan algoritma yang meningkatkan komputer untuk berkembang lebih jauh berdasarkan pengalaman dari data-data yang dimasukkan. *Machine learning* memerlukan data untuk belajar, sehingga terdapat istilah *learn from data*. Input data-data yang dimasukkan ke *machine learning* akan membuat sebuah algoritma yang menghasilkan suatu model. Dari model tersebut maka dihasilkanlah suatu prediksi ataupun pengambilan keputusan yang didasarkan dari data-data yang sudah dimasukkan sebelumnya.

Salah satu tugas dari *machine learning* ialah klasifikasi, untuk melakukan proses klasifikasi maka semua informasi yang dibutuhkan harus dikumpulkan yang mana pada akhirnya akan menghasilkan *output* berupa jenis klasifikasi yang kita inginkan, salah satu contohnya adalah untuk meramal cuaca pada hari tertentu apakah akan cerah, berawan, ataupun hujan.

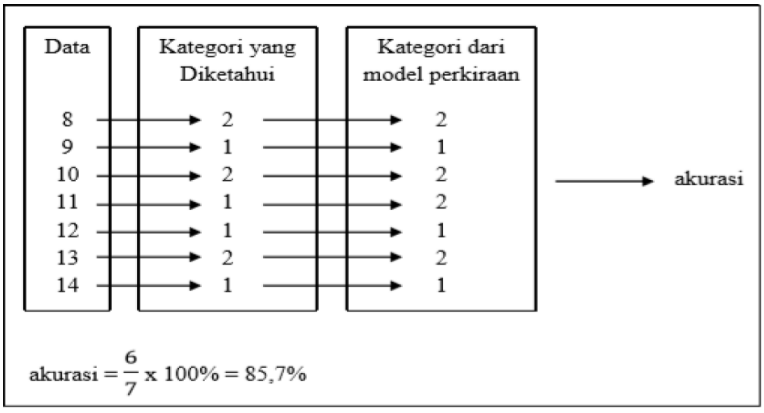
## 2.5 Klasifikasi

Menurut Han dan Kamber (2006) klasifikasi adalah proses menentukan suatu model atau fungsi yang dapat menjelaskan kelas data sehingga data yang belum memiliki label dapat ditentukan kategorinya. Klasifikasi terdiri dari dua tahap yaitu tahap *learning* dan tahap *testing*. Pada gambar 3.1 adalah contoh tahap *learning* yang mana data-data yang ada dikelompokkan berdasarkan kategori yang sudah diketahui dan selanjutnya akan membentuk model perkiraan.




**Gambar 3. 1** Tahap *Learning* (Sumber: Pratiwi, 2017)

Selanjutnya pada gambar 3.2 adalah contoh tahap *testing* yang mana model yang sudah terbentuk sebelumnya akan dilakukan pengujian terhadap data-data lainnya (selain data yang telah digunakan pada tahap *learning*) sehingga akan didapatkan nilai akurasi model yang dihasilkan berdasarkan pada tahap learning sebelumnya. Apabila nilai akurasi model dirasa sudah cukup maka model ini akan digunakan untuk memprediksi data lainnya diluar dari data yang telah digunakan.




**Gambar 3. 2** Tahap *Testing* (Sumber: Pratiwi, 2017)

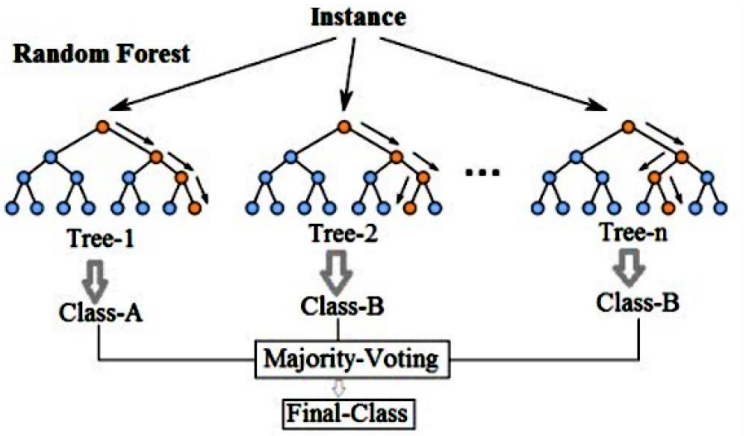
## 2.6 *Random Forest*

Menurut Wezel dan Potharst (2007) konsep *random forest* pertama kali dikemukakan oleh Tin Kam Ho pada tahun 1995, kemudian dikembangkan oleh Breiman pada tahun 2001. *Random forest* adalah salah satu metode untuk meningkatkan akurasi suatu klasifikasi data dari sebuah pengelompokan tunggal yang tidak stabil menggunakan kombinasi banyak pengelompokan dari suatu metode yang sama dengan proses *voting* untuk mendapatkan prediksi klasifikasi akhir.

Metode *random forest* merupakan pengembangan dari metode *Classification and Regression Tree* (CART) dengan menerapkan metode *bootstrap aggregating* (*bagging*) dan *random feature selection*. CART adalah metode eksplorasi data yang didasarkan pada teknik pohon keputusan yang mana pohon klasifikasi akan dihasilkan jika variabel dependen yang digunakan bersifat kategorik dan akan menghasilkan pohon regresi jika variabel dependen yang digunakan bersifat numerik (Breiman, 2001).

Menurut Liaw dan Wiener (2002) *bootstrap aggregating* (*bagging*) merupakan metode yang mampu digunakan untuk membentuk sampel *bootstrap* dari data kandidat atribut untuk dibagi pada setiap *node* yang berasal dari himpunan atribut acak dari hasil data yang digunakan. *Random forest* melakukan proses pengacakan tidak hanya pada data sampel yang digunakan melainkan juga dilakukan pada pemilihan variabel independennya sehingga pohon klasifikasi yang dibangkitkan akan memiliki bentuk dan ukuran yang berbeda-beda.

Menurut Mambang dan Byna (2017) *random forest* adalah klasifikasi yang terdiri dari beberapa pohon keputusan yang dibentuk dengan mengunakan vektor acak. *Random forest* adalah pengembangan dari pohon keputusan dengan menggunakan beberapa pohon keputusan dimana pada setiap pohon keputusan dilakukan proses *training* data menggunakan sampel individu dan setiap atribut dipecah pada pohon yang terpilih antara atribut *subset* yang bersifat acak dan hasil klasifikasi dipilih berdasarkan jumlah suara terbanyak (nilai yang paling banyak muncul) berdasarkan hasil dari kumpulan semua pohon keputusan.




**Gambar 3. 3** Contoh Algoritma *Random Forest*
(Sumber: Koehrsen, 2017)

Menurut Breiman (2001) dalam metode *random forest* banyak pohon yang ditumbuhkan sehingga membentuk hutan (*forest*) yang akan dianalisis. Pada gugus data yang terdiri atas *n* (pengamatan) dan *p* (variabel independen) *random forest* dilakukan dengan cara berikut yaitu:

1. Melakukan penarikan contoh acak berukuran *n* dengan pemulihan pada gugus data yang mana tahapan ini disebut tahapan *bootstrap*.
2. Dengan menggunakan contoh *bootstrap*, pohon dibangun sampai mencapai ukuran maksimal tanpa pemangkasan. Pada setiap simpul dilakukan pemilihan terhadap *p* (variabel independen) dalam jumlah tertentu sebanyak *m* yang dilakukan secara acak dimana nilai *m* lebih kecil dari nilai *p*, tahapan ini disebut dengan tahapan *random feature selection*.
3. Ulangi langkah 1 dan 2 sebanyak *k* kali, sehingga terbentuklah sebuah hutan yang terdiri atas *k* pohon.

Metode *random forest* harus menentukan *m* (jumlah variabel prediktor) yang diambil secara acak dan *k* pohon yang akan digunakan agar mendapatkan hasil yang optimum. Ukuran *m* sangat mempengaruhi korelasi dan kekuatan masing-masing pohon. Untuk menentukan nilai *m* yang diambil secara acak terhadap *p* (variabel independen) maka cara menentukannya adalah sebagai berikut (Hastie dkk, 2008):

1. Untuk klasifikasi, penentuan nilai *m* adalah dengan cara $\sqrt{p}$ dengan nilai node atau simpul kecil adalah 1.

2. Untuk regresi, penentuan nilai *m* adalah dengan cara $^{p}/_{3}$ dengan nilai node atau simpul kecil adalah 5.

Sedangkan menurut Breiman dan Cutler (2003) terdapat tiga cara untuk memperoleh nilai *m* untuk mengetahui *error* OOB yaitu:

$$m = \frac{1}{2}\sqrt{p} \tag{3.1}$$

$$m = \sqrt{p} \tag{3.2}$$

$$m = 2 \times \sqrt{p} \tag{3.3}$$

yang mana *p* = jumlah variabel independen.

Menurut Breiman (2001) penggunaan nilai *m* yang tepat akan menghasilkan *random forest* dengan hubungan antar pohon yang lumayan kecil tetapi memiliki kekuatan setiap pohon yang cukup besar yang ditunjukkan dengan perolehan nilai *error* OOB bernilai kecil. OOB (*out of bag*) merupakan nilai pengamatan dari gugus data asli yang tidak termuat dalam contoh *bootstrap* pada setiap iterasinya. Data OOB tidak digunakan untuk membangun pohon melainkan digunakan sebagai data validasi pada pohon yang bersesuaian. Nilai salah klasifikasi *random forest* diduga melalui *error* OOB yang diperoleh dengan cara yaitu:

1. Lakukan prediksi terhadap setiap data OOB pada pohon yang bersesuaian.
2. Umumnya setiap amatan gugus data asli akan menjadi data OOB sebanyak sekitar 36% atau sepertiga dari jumlah pohon yang dibentuk. Oleh sebab itu pada langkah 1 tiap-tiap amatan gugus data asli mengalami prediksi sebanyak sekitar sepertiga kali dari banyaknya pohon. Jika *x* adalah sebuah amatan dari gugus data asli maka hasil prediksi *random forest* terhadap *x* merupakan gabungan dari hasil prediksi setiap kali *x* menjadi data OOB.
3. *Error* OOB bergantung pada korelasi antar pohon dan kekuatan masing-masing pohon dalam *random forest* yang mana peningkatan korelasi dapat meningkatkan nilai *error* OOB sedangkan penambahan jumlah pohon yang dibentuk akan menurunkan nilai *error* OOB. *Error* OOB dihitung berdasarkan proporsi kesalahan klasifikasi hasil prediksi *random forest* dari seluruh jumlah amatan gugus data asli. Penggunaan banyak pohon akan menghasilkan *variabel importance* yang semakin stabil.

### 2.6.1 Ukuran Ketepatan Klasifikasi

Menurut Bramer (2007) kinerja klasifikasi dapat diukur dengan menggunakan *confusion matrix* yang memberikan keputusan yang diperoleh dalam pelatihan dan pengujian. *Confusion matrix* adalah alat yang berfungsi untuk menganalisis seberapa baik penggolongan dapat mengenali jenis dari kelas yang berbeda dimana kelas yang diprediksi akan ditampilkan dibagian kiri matriks dan kelas yang diobservasi ditampilkan dibagian atas (Han dan Kamber, 2006). Berikut ini adalah bentuk tabel *confusion matrix* seperti yang ada pada tabel 3.1.

**Tabel 3. 1** *Confusion Matrix*

| | ***Actual Positive Class*** | ***Actual Negative Class*** |
|---|---|---|
| ***Predicted Positive Class*** | TP *(True Positive)* | FP *(False Positive)* |
| ***Predicted Negative Class*** | FN *(False Negative)* | TN *(True Negative)* |

Dimana:

TP (*True Positive*) : Jumlah prediksi yang tepat bersifat positif.

FP (*False Positive*) : Jumlah prediksi yang salah bersifat positif.

FN (*False Negative*) : Jumlah prediksi yang salah bersifat negatif.

TN (*True Negative*) : Jumlah prediksi yang tepat bersifat negatif.

Menurut Bramer (2013) *False Positive* disebut dengan *error* tipe satu yang mana hal ini terjadi ketika kasus yang harusnya diklasifikasikan sebagai negatif tapi diklasifikasikan positif, sedangkan *False Negative* disebut dengan *error* tipe dua yang terjadi ketika kasus yang harusnya diklasifikasikan sebagai positif tapi diklasifikasikan negatif.

Menurut Sembiring (2007) *dataset* yang memiliki jumlah kelas negatif (kelas mayoritas) jauh lebih banyak daripada jumlah kelas positif (kelas minoritas) maka disebut dengan *imbalanced dataset*. Dalam menganalisa data yang tidak seimbang tingkat akurasi klasifikasi secara umum seringkali tidak bisa dijadikan landasan dalam penentuan ukuran kinerja yang tepat. Tingkat akurasi cenderung dominan pada ketepatan data kelas minoritas sehingga acuan yang tepat adalah dengan mengamati beberapa matriks diantara yaitu AUC (*Area Under the ROC Curve*), *G-mean*, *Apparent Error Rate* (APER), dan *Total Accuracy Rate* (1-APER) (Zhang & Wang, 2011).

Berikut ini merupakan beberapa rumus dari beberapa matriks yang dibisa dijadikan acuan dalam menentukan tingkat kinerja pada analisis data yang tidak seimbang (Wang dan Yao, 2013):

*True Positive Rate / Sensitivity / Recall* $= \frac{TP}{TP+FN}$ (3.4)

*True Negative Rate / Specificity* $= \frac{TN}{TN+FP}$ (3.5)

*Precision* atau PPV $= \frac{TP}{TP+FP}$ (3.6)

*Apparent Error Rate* (APER) $= \frac{FP+FN}{N}$ (3.7)

*Total Accuracy Rate* (1-APER) $= \frac{TP+TN}{N}$ (3.8)

*G-mean* $= \sqrt{Recall \times Specificity}$ (3.9)

*False Positive Rate* $= \frac{FP}{FP+TN}$ (3.10)

*Area Under the ROC Curve* $= \frac{1+Recall-FP\ Rate}{2}$ (3.11)

*Accuracy* $= \frac{TP+TN}{TP+TN+FP+FN}$ (3.12)

Menurut Prasetio dan Pratiwi (2015) *Sensitivity* dan *Specificity* berfungsi sebagai ukuran statistik dari kinerja klasifikasi *biner*, mengukur model terbaik, dan memilih model yang paling efisien. *Sensitivity* mengukur proporsi *True Positive* yang diidentifikasi dengan benar, sedangkan *Specificity* mengukur proporsi *True Negative* yang diidentifikasi dengan benar. *Precision* adalah tingkat ketepatan dan ketelitian hasil pengamatan. Nilai *G-mean* berguna untuk melihat tingkat keseimbangan akurasi prediksi yang biasa digunakan pada analisis data yang tidak seimbang. Menurut Jatmiko dkk (2017) hal ini diperlukan karena metode klasifikasi cenderung baik dalam memprediksi kelas dengan data sampel yang banyak namun buruk dalam memprediksi kelas dengan data sampel yang sedikit.

*Area Under the ROC Curve* (AUC) adalah suatu ukuran numerik untuk membedakan kinerja model dan menunjukkan seberapa bagus peringkat model dengan memisahkan pengamatan positif dan pengamatan negatif (Attenberg dan Ertekin, 2013). Dengan membandingkan dengan pengklasifikasian yang lain, AUC adalah cara yang baik untuk mendapatkan nilai kinerja pengklasifikasian secara

umum (Japkowicz, 2000). AUC merupakan kinerja yang populer dalam ketidakseimbangan kelas dimana semakin tinggi nilai AUC yang ditunjukkan dari hasil kinerja yang didapat maka semakin bagus pula model yang dihasilkan (Liu dan Zhou, 2013).

### 2.6.2 Ukuran Tingkat Kepentingan

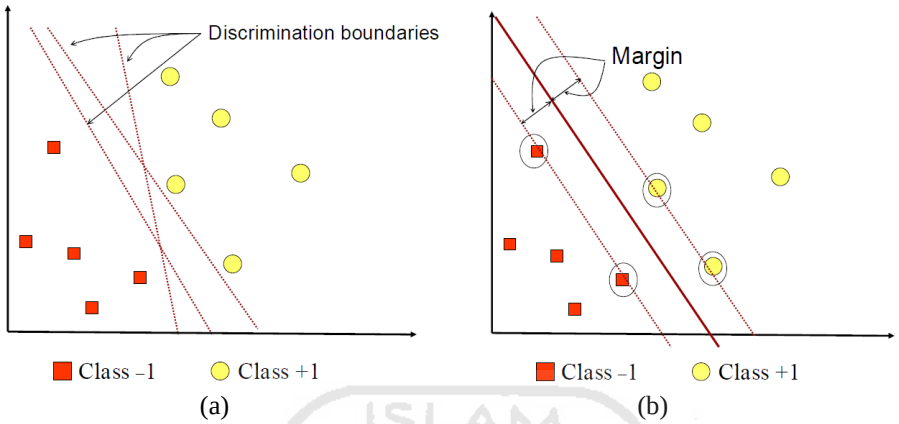
*Mean Decrease Accuracy* (MDA) adalah salah satu ukuran tingkat kepentingan (*variable importance*) variabel independen yang dihasilkan oleh metode *random forest*. MDA menampilkan seberapa besar tambahan observasi yang mengalami kesalahan klasifikasi jika salah satu variabel independen tidak diikutsertakan ke dalam pengujian. Ukuran tingkat kepentingan lainnya yaitu *Mean Decrease Gini* (MDG), ukuran tersebut digunakan untuk mengetahui tingkat kestabilan tiap variabel independen dalam *random forest*. Semakin besar nilainya maka akan semakin baik (Breiman, 2000).

## 2.7 *Support Vector Machine* (SVM)

Pada bidang *pattern recognition* metode SVM merupakan rangkaian konsep unggulan yang diperkenalkan pertama kali oleh Vapnik pada tahun 1992. Meski usia SVM masih terbilang muda, saat ini SVM merupakan salah satu metode yang berkembang pesat. SVM bertujuan untuk menemukan *hyperlane* terbaik untuk memisahkan dua kelas data dan merupakan salah satu metode *machine learning* yang bekerja atas prinsip *Structural Risk Minimization* (SRM). SVM bekerja dengan memaksimalkan margin yang merupakan jarak pemisah antara kedua kelas data. Meskipun pada dasarnya SVM memiliki prinsip linier, akan tetapi SVM telah berkembang sehingga dapat bekerja pada masalah *non*-linier. Cara kerja SVM pada permasalahan *non*-linier adalah dengan memasukkan konsep *kernel* pada ruang berdimensi tinggi. Pada ruang berdimensi ini nantinya akan dicari pemisah atau *hyperlane*. *Hyperlane* terbaik dapat ditentukan dengan mengukur margin dan mencari titik maksimalnya dan usaha ini merupakan inti dari proses pada metode SVM (Pratiwi, 2017).

### 2.7.1 *Linear Separable Data*

Metode SVM dengan *hyperlane* yang berbentuk garis lurus disebut dengan *linear separable*. Adapun contohnya dapat dilihat pada gambar 3.4.

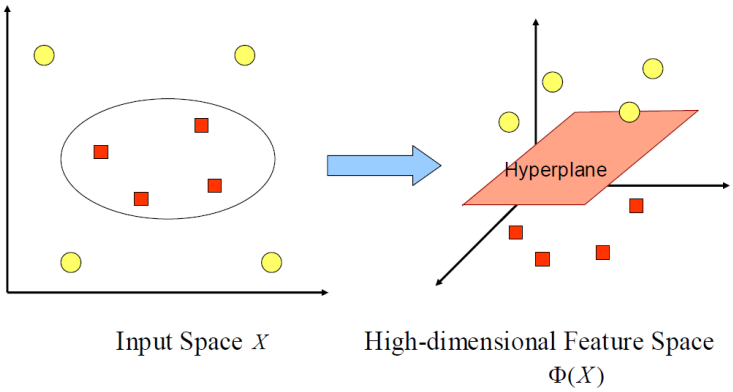



**Gambar 3. 4** Garis Pemisah Linier (Sumber: Nugroho, 2003)

Pada gambar 3.4 terdapat beberapa pola data yang terdiri dari dua kelas yaitu kelas -1 berwarna merah dan kelas +1 berwarna kuning, pada bagian (a) terdapat beberapa macam garis pemisah (*hyperlane*) alternatif yang dapat digunakan sedangkan pada bagian (b) didapatkanlah *hyperlane* terbaik yang berada tepat ditengah-tengah antar kelas, pola data yang dilingkari merupakan pola data terdekat dari *hyperlane* yang disebut dengan *support vector* yang berguna untuk membantu menemukan bentuk *hyperlane* terbaik. Pada gambar 3.4 dapat dilihat bahwa dibutuhkan empat buah *support vector* dalam menentukan *hyperlane* terbaik.

### 2.7.2 *Non-Linear Separable Data*

Pada permasalahan di dunia nyata pada umumnya data yang diperoleh jarang sekali yang bersifat *linear* dan justru banyak yang bersifat *non-linear*. Pada metode SVM terdapat suatu fungsi *kernel* yang berfungsi untuk menyelesaikan permasalahan *non-linear*. *Kernel* berfungsi memungkinkan suatu model diimplementasikan pada ruang dimensi lebih tinggi. Dalam *non-linear* SVM, data digambarkan oleh suatu fungsi ruang vektor yang berdimensi lebih tinggi. Pada ruang vektor yang baru ini *hyperlane* yang memisahkan kedua kelas tersebut dapat dikonstruksikan (Nugroho, 2003).




**Gambar 3. 5** Ilustrasi Data *Non-Linear* Pada *SVM*

(Sumber: Nugroho, 2003)

Pada gambar 3.5 diperlihatkan bahwa data pada kelas kuning dan data pada kelas merah yang berada pada ruang input berdimensi dua tidak dapat dipisahkan secara linier. Oleh karenanya dilakukan pemetaan ruang input ke ruang vektor baru yang berdimensi lebih tinggi (tiga dimensi) dimana kedua kelas data dapat dipisahkan secara linier oleh sebuah *hyperlane*. Secara matematis beberapa fungsi *kernel* yang umum dipakai dalam dalam SVM dirumuskan sebagai berikut ini (Nugroho, 2003):

*Kernel Linear*: $K(\vec{x}_i, \vec{x}_j) = \vec{x}_i{}^T \vec{x}_j$ (3.13)

*Kernel Radial*: $K(\vec{x}_i, \vec{x}_j) = \exp\left(-\frac{1}{2\sigma^2}\left\|\vec{x}_i - \vec{x}_j\right\|^2\right)$ (3.14)

*Kernel Polynomial*: $K(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \vec{x}_j + 1)^P$ (3.15)

*Kernel Sigmoid*: $K(\vec{x}_i, \vec{x}_j) = \tanh(\alpha \vec{x}_i \vec{x}_j + \beta)$ (3.16)

## 2.8 *Imbalanced Data Sampling Method*

*Imbalanced data* adalah suatu kondisi dimana terjadi ketidakseimbangan antara jumlah data antar kelas yang berbeda, salah satu kelasnya memiliki jumlah data yang lebih banyak (kelas mayoritas) sedangkan kelas lainnya memiliki jumlah data yang lebih kecil (kelas minoritas). *Balancing data* adalah mengubah data yang tidak seimbang menjadi seimbang, metode ini secara umum disebut dengan metode sampling. Umumnya metode ini berfungsi untuk memodifikasi data yang tidak seimbang ke dalam distribusi yang lebih seimbang menggunakan beberapa cara

dengan mengubah ukuran kumpulan data asli dan memberikan proporsi keseimbangan yang sama (Sastrawan dkk, 2010).

Hasil pengujian metode klasifikasi pada data yang tidak seimbang biasanya memiliki ciri khas yaitu tingkat kesalahan klasifikasi di data kelas minoritas lebih tinggi dibandingkan dengan tingkat kesalahan klasifikasi di data kelas mayoritas. Untuk mengatasi permasalahan tersebut dapat dilakukan sampel ulang (*re-sampling*) data asli. Beberapa teknik sampling untuk mengatasi *imbalanced data* pada *machine learning* diantaranya yaitu (Japkowicz, 2000):

1. *Undersampling*

   Teknik *undersampling* adalah metode sampling yang melakukan *re-sampling* pada data kelas mayoritas dengan cara mengeliminasi data kelas mayoritas secara acak hingga jumlahnya sebanyak data kelas minoritas.

2. *Oversampling*

   Teknik *oversampling* adalah metode sampling yang melakukan *re-sampling* pada data kelas minoritas dengan cara menambah data kelas minoritas secara acak hingga jumlahnya sebanyak data kelas mayoritas.

# BAB IV
# METODOLOGI PENELITIAN

## 3.1 Populasi dan Sampel Penelitian

Populasi dalam penelitian ini adalah hasil pertandingan sepak bola Liga 1 Indonesia sedangkan sampelnya adalah hasil pertandingan musim kompetisi 2018.

## 3.2 Variabel Penelitian

Variabel penelitian yang digunakan pada penelitian ini terbagi dua yaitu:

1. Variabel prediktor atau variabel independen, yang terdiri dari total tembakan, tembakan *on target*, akurasi operan, tekel sukses, sepak pojok, pelanggaran, *offside*, kartu kuning, dan kartu merah.
2. Variabel respon atau variabel dependen, yaitu hasil pertandingan.

**Tabel 4. 1** Definisi Operasional Variabel

| No | Variabel | Definisi |
|---|---|---|
| 1 | Total Tembakan (X1) | Semua percobaan tendangan baik yang mengarah ke gawang (tembakan *on target*) ataupun yang melenceng, termasuk yang terkena tiang gawang. |
| 2 | Tembakan O*n Target* (X2) | Semua percobaan tendangan yang mengarah ke gawang, baik bola itu masuk ke gawang, berhasil dihalau kiper, pemain belakang atau pemain terakhir lawan. |
| 3 | Akurasi Operan (X3) | Tingkat keberhasilan dalam proses memberikan atau mengumpan bola pada rekan setim baik itu umpan panjang, umpan pendek, atau umpan belakang ke kiper. |
| 4 | Tekel Sukses (X4) | Bola berhasil direbut oleh pemain yang melakukan tekel atau salah satu rekan tim |

| | | |
|---|---|---|
| | | atau bola keluar dari arena permainan dan menjadi "aman". Tekel dikatakan gagal ketika usaha tekel yang dilakukan tidak berhasil atau berujung pada pelanggaran. |
| 5 | Sepak Pojok (X5) | Tendangan dari daerah sudut lapangan karena bola melewati garis gawang setelah menyentuh pemain bertahan. |
| 6 | Pelanggaran (X6) | Suatu tindakan yang menyalahi aturan yang ditetapkan oleh wasit. |
| 7 | *Offside* (X7) | Pemain penyerang atau rekan tim paling depan berada pada posisi lebih dekat ke gawang lawan daripada pemain bertahan lawan sebelum bola diumpan oleh rekannya. |
| 8 | Kartu Kuning (X8) | Peringatan akibat terjadinya pelanggaran ringan, didalam sebuah pertandingan sepak bola pemain hanya boleh menerima satu kartu kuning. Lebih dari itu akan dikenakan kartu merah oleh wasit. |
| 9 | Kartu Merah (X9) | Hukuman bagi pemain yang melakukan kesalahan kategori berat seperti berkata-kata kasar dan tidak sopan pada wasit, bermain curang, dan kedapatan sengaja ingin mencederai pemain lain. Konsekuensi dari diberikannya kartu merah yaitu pemain harus keluar dari arena permainan. |
| 10 | Hasil Pertandingan (Y) | Hasil pertandingan pada penelitian ini dikelompokkan menjadi tiga yaitu hasil menang, hasil seri dan hasil kalah). |

### 3.3 Jenis dan Sumber Data

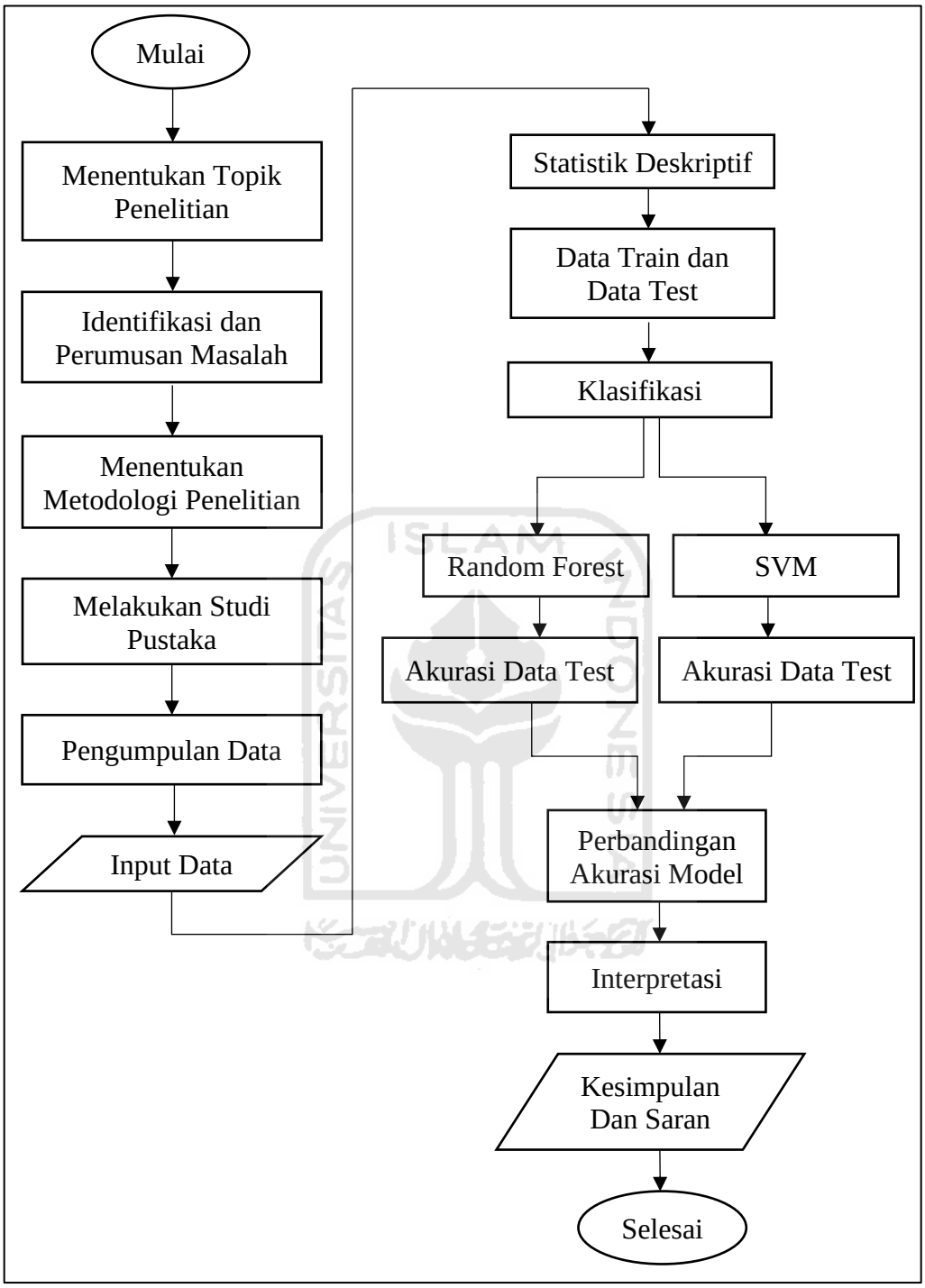
Jenis data yang digunakan dalam penelitian ini adalah data sekunder. Data tersebut diambil dari situs resmi Liga 1 Indonesia selaku kompetisi sepak bola tertinggi di Indonesia yang menampilkan hasil statistik setiap pertandingan pada musim kompetisi 2018 yang diikuti oleh 18 tim sepak bola terbaik dari seluruh wilayah Indonesia.

### 3.4 Metode Analisis Data

*Software* yang digunakan dalam penelitian ini adalah *Microsoft Excel* 2019 dan *R Studio* 1.2.1335. Metode yang digunakan dalam penelitian ini adalah metode analisis deskriptif untuk mengetahui gambaran umum dari data yang diteliti serta penggunaan metode *random forest* dan *support vector machine* untuk dilakukan perbandingan hasil analisis klasifikasi.

### 3.5 Tahapan Penelitian

1. Pada penelitian ini tahapan awal dimulai dari penentuan topik yang akan dipilih.
2. Setelah itu melakukan identifikasi dan perumusan masalah yang akan diteliti pada penelitian ini.
3. Selanjutnya dilakukan penentuan metodologi dan studi pustaka terhadap penelitian-penelitian sebelumnya.
4. Pengumpulan atau pengambilan data sekunder dan kemudian melakukan pembersihan data sehingga data dapat digunakan dengan baik dalam penelitian.
5. Melakukan analisis deskriptif pada data yang diteliti.
6. Peneliti melakukan pembagian data menjadi data *training* dan data *testing* untuk digunakan dalam permodelan *machine learning.*
7. Pada tahap klasifikasi dibagi menjadi 2 metode yaitu *Random Forest* dan *Support Vector Machine.* Setelah terbentuk model maka dilakukan perhitungan akurasi data *testing*.
8. Setelah itu melakukan perbandingan akurasi antar model yang kemudian di lakukan interpretasi dan didapat kesimpulan dari hasil penelitian.




**Gambar 4. 1** Diagram Alir Penelitian

# BAB V
# HASIL DAN PEMBAHASAN

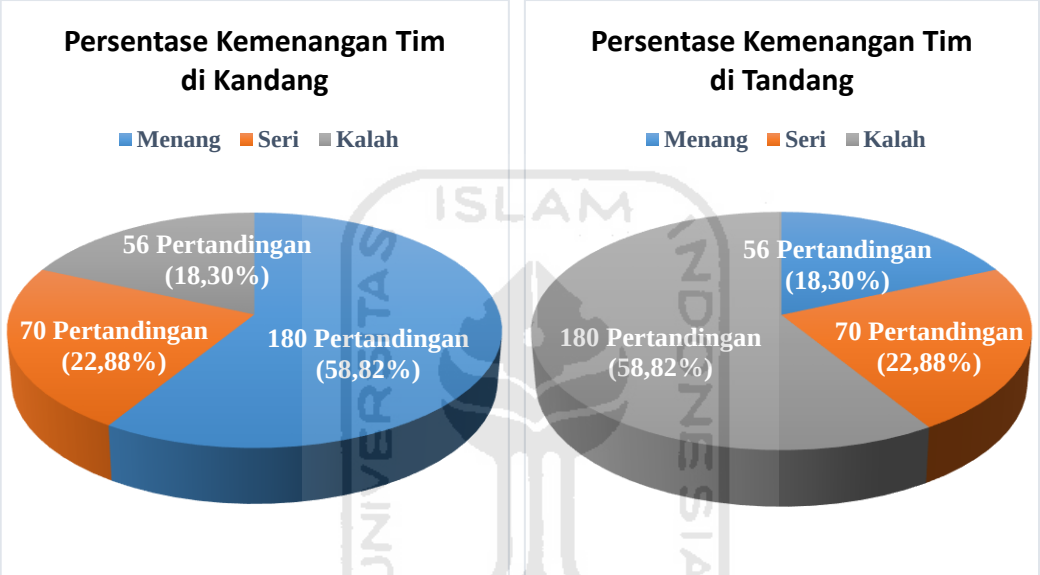
## 4.1 Analisis Deskriptif

Liga 1 Indonesia musim kompetisi 2018 diselenggarakan pada tanggal 3 Maret sampai 9 Desember 2018 dan diikuti oleh 18 tim dari berbagai daerah di Indonesia. Setiap tim memainkan 34 pertandingan dengan rincian 17 pertandingan dimainkan di kandang sendiri dan 17 pertandingan lainnya dimainkan di kandang lawan (Pertandingan Tandang). Tim Persija Jakarta berhasil keluar sebagai juara pada musim ini dengan mengumpulkan 62 poin hasil dari 18 kemenangan, 8 seri, dan 8 kekalahan. Sementara itu tiga tim dengan poin terendah yaitu Mitra Kukar FC, Sriwijaya FC, dan PSMS Medan akan terlempar ke kompetisi Liga 2 musim selanjutnya dan digantikan oleh 3 tim teratas dari kompetisi Liga 2 musim 2018.

**Tabel 5. 1** Klasemen akhir Liga 1 2018

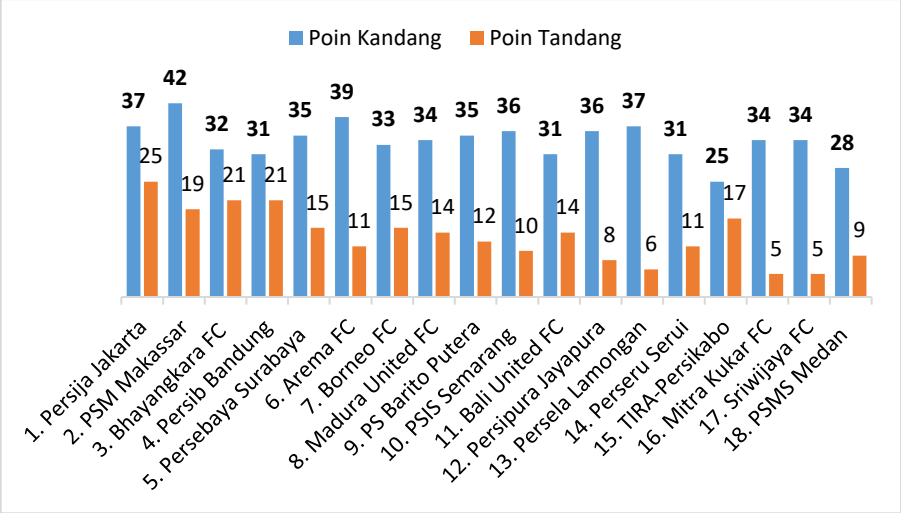
| Posisi | Klub | Pertandingan | Menang | Seri | Kalah | Poin |
|---|---|---|---|---|---|---|
| 1 | Persija Jakarta | 34 | 18 | 8 | 8 | 62 |
| 2 | PSM Makassar | 34 | 17 | 10 | 7 | 61 |
| 3 | Bhayangkara FC | 34 | 15 | 8 | 11 | 53 |
| 4 | Persib Bandung | 34 | 14 | 10 | 10 | 52 |
| 5 | Persebaya Surabaya | 34 | 14 | 8 | 12 | 50 |
| 6 | Arema FC | 34 | 14 | 8 | 12 | 50 |
| 7 | Borneo FC | 34 | 14 | 6 | 14 | 48 |
| 8 | Madura United FC | 34 | 13 | 9 | 12 | 48 |
| 9 | PS Barito Putera | 34 | 12 | 11 | 11 | 47 |
| 10 | PSIS Semarang | 34 | 13 | 7 | 14 | 46 |
| 11 | Bali United FC | 34 | 12 | 9 | 13 | 45 |
| 12 | Persipura Jayapura | 34 | 12 | 8 | 14 | 44 |
| 13 | Persela Lamongan | 34 | 11 | 10 | 13 | 43 |
| 14 | Perseru Serui | 34 | 11 | 9 | 14 | 42 |
| 15 | TIRA-Persikabo | 34 | 12 | 6 | 16 | 42 |
| 16 | Mitra Kukar FC | 34 | 12 | 3 | 19 | 39 |
| 17 | Sriwijaya FC | 34 | 11 | 6 | 17 | 39 |
| 18 | PSMS Medan | 34 | 11 | 4 | 19 | 37 |

Dari 306 total pertandingan di Liga 1 Indonesia musim 2018, tim yang bertanding di kandang sendiri meraih kemenangan sebanyak 180 pertandingan, lalu 70 pertandingan berakhir seri dan 56 pertandingan berakhir dengan kekalahan, lalu sebaliknya tim yang bertanding di kendang lawan (tandang) meraih kemenangan sebanyak 56 pertandingan, lalu 70 pertandingan berakhir seri dan 180 pertandingan berakhir dengan kekalahan, perbandingannya dapat digambarkan secara visual seperti grafik pada Gambar 5.1.




**Gambar 5. 1** Perbandingan Pertandingan Kandang dan Tandang

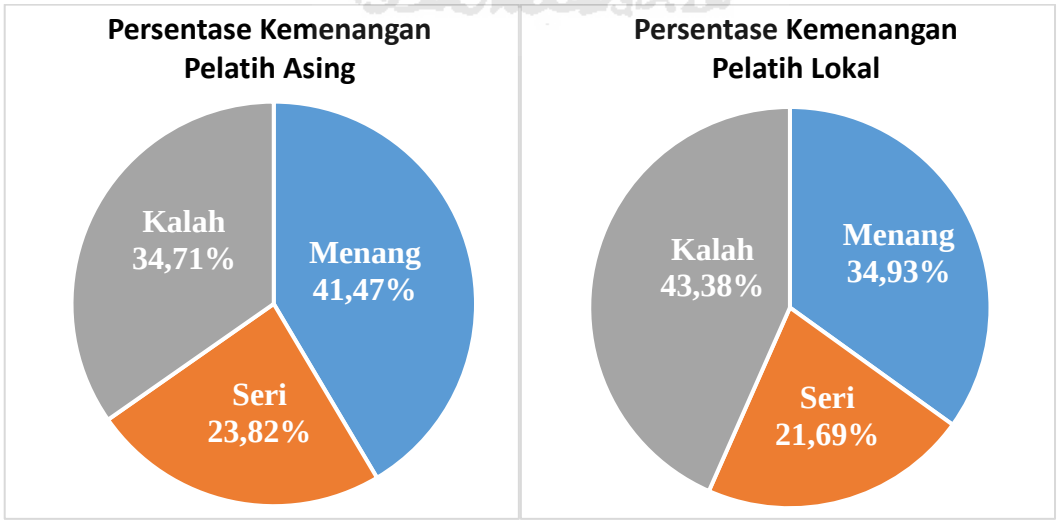
Selanjutnya untuk melihat persebaran poin kandang dan tandang tiap tim yang bertanding pada Liga 1 Indonesia musim 2018 dapat dilihat pada gambar 5.2.




**Gambar 5. 2** Persebaran Poin Kandang dan Tandang Tiap Tim

Pada gambar 5.2 memberikan informasi bahwa dalam memperoleh poin di kandang sendiri tim PSM Makassar adalah tim yang memiliki poin paling banyak (42 poin) lalu disusul oleh tim Arema FC (39 poin), Persija Jakarta dan Persela Lamongan (37 poin). Sedangkan tim yang paling banyak memperoleh poin pada pertandingan tandang adalah tim Persija Jakarta (25 poin), Bhayangkara FC dan Persib Bandung (21 poin). Secara keseluruhan setiap tim memiliki jumlah poin kandang lebih banyak dibandingkan poin tandang dikarenakan peluang tim dalam memenangkan pertandingan lebih besar jika bertanding di kandang sendiri.

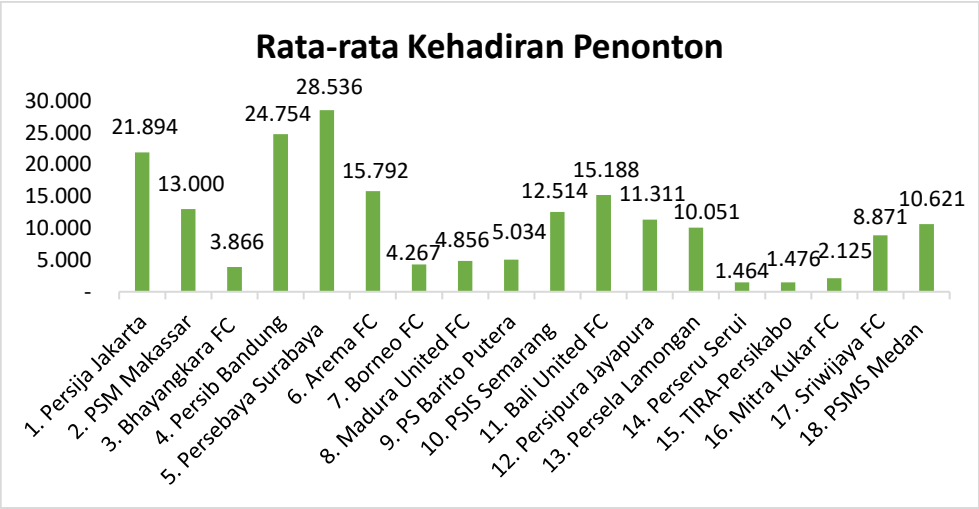
Setiap tim pada Liga 1 mempunyai seorang pelatih yang dibedakan menjadi dua jenis yaitu pelatih asing yang merupakan pelatih yang memiliki kewarganegaraan selain Indonesia dan pelatih lokal yang merupakan pelatih yang memiliki kewarganegaraan Indonesia. Jika dilihat berdasarkan perbedaan kewarganegaraan tersebut, tim yang memiliki pelatih asing mempunyai persentase hasil kemenangan lebih besar dibandingkan pelatih lokal yang mana pelatih asing memiliki persentase hasil kemenangan sebesar 41,47%, hasil seri sebesar 23,82%, dan hasil kalah sebesar 34,71% sedangkan pelatih lokal memiliki persentase hasil kemenangan sebesar 34,93%, hasil seri sebesar 21,69%, dan hasil kalah sebesar 43,38%. Untuk melihat perbandingan persentase hasil kemenangan antara pelatih asing dan lokal di Liga 1 Indonesia musim 2018 dapat dilihat pada gambar 5.3.




**Gambar 5. 3** Perbandingan Persentase Kemenangan Pelatih Asing dan Lokal

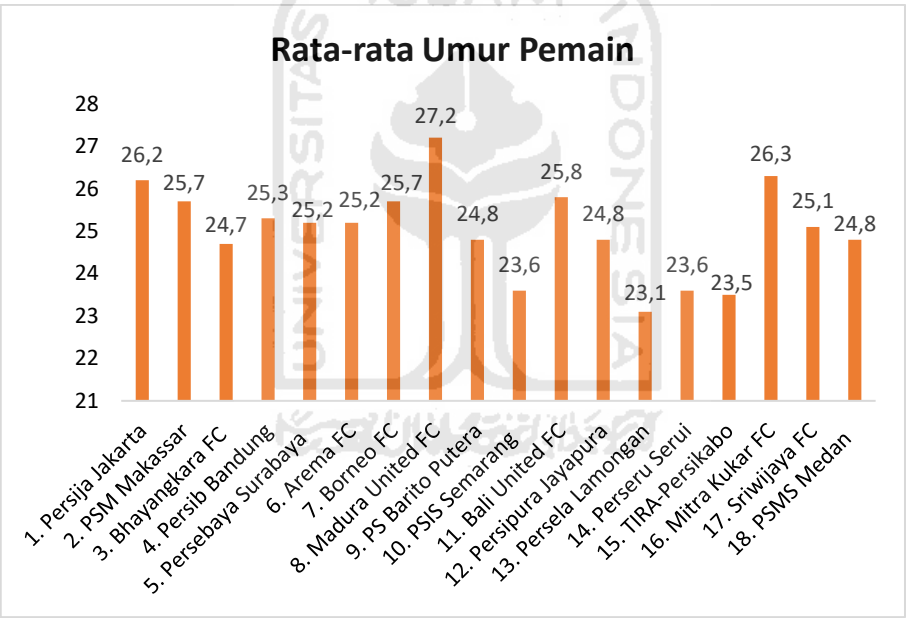
Dari gambar 5.3 diatas dapat kita tarik kesimpulan bahwa kualitas pelatih asing lebih bagus dibandingkan pelatih lokal, hal ini tentunya menjadi PR besar bagi pelatih-pelatih lokal di Indonesia untuk lebih meningkatkan kualitas diri agar tidak ketinggalan jauh dari pelatih asing. Dalam hal ini pelatih lokal bisa untuk meningkatkan lisensi kepelatihan mereka agar kualitas pelatihan mereka menjadi meningkat sehingga kualitas kompetisi ligapun akan menjadi lebih baik.

Setiap pertandingan sepakbola tentu akan lebih meriah dan menegangkan jika disaksikan oleh banyak pasang mata, terlebih lagi jika terdapat banyak penonton di stadion yang langsung memberikan dukungan pada tim kesayangan mereka, tim-tim yang berlaga di Liga 1 Indonesia memiliki jumlah penonton yang beragam dan pada musim 2018 rata-rata kehadiran penonton terbanyak dimiliki oleh tim Persebaya Surabaya yang memiliki rata-rata kehadiran penonton sebanyak 28.536 penonton disetiap pertandingan ketika Persebaya bermain di kandang. Lalu setelahnya di posisi kedua ada Persib Bandung dengan jumlah 24.754 penonton dan Persija Jakarta di posisi ketiga dengan jumlah 21.894 penonton. Jumlah penonton yang banyak dimiliki oleh tiga klub tersebut dikarena mereka memiliki suporter fanatik dalam jumlah yang banyak serta ditunjang pula dengan kapasitas stadion dalam menampung penonton yang lebih besar dibandingkan stadion tim-tim lainnya yang berkompetisi di liga 1 Indonesia. Untuk melihat perbandingan selengkapnya dapat dilihat pada gambar 5.4.



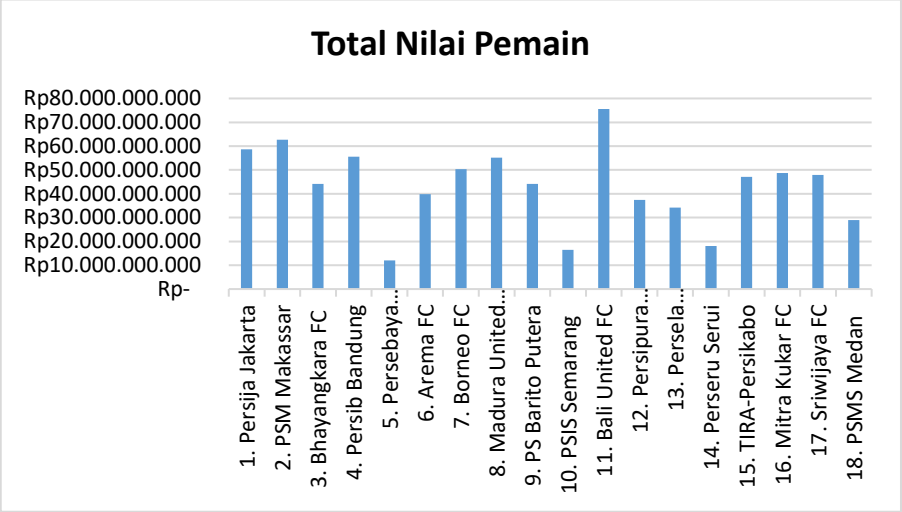

**Gambar 5. 4** Persebaran Rata-rata Jumlah Kehadiran Penonton

Setiap tim di Liga 1 Indonesia memiliki komposisi pemain yang berbeda-beda, pemilihan pemain senior dan pemain junior tentu akan memberikan pengaruh bagi performa tim dalam setiap pertandingan. Ada beberapa tim yang memainkan pemain senior lebih banyak untuk memanfaatkan pengalaman mereka membantu tim memenagkan pertandingan, dan ada pula beberapa tim yang memainkan pemain junior lebih banyak agar dikemudian hari pemain dapat lebih berkembang dan menjadi aset penting bagi tim. Pada Liga 1 Indonesia musim 2018, tim yang memiliki rata-rata umur pemain paling tinggi adalah Madura United FC dengan rata-rata umur pemain 27,2 tahun sedangkan tim yang memiliki rata-rata umur pemain paling rendah adalah Persela Lamongan dengan rata-rata umur pemain 23,1 tahun. Untuk melihat lebih lengkapnya dapat dilihat pada gambar 5.5.



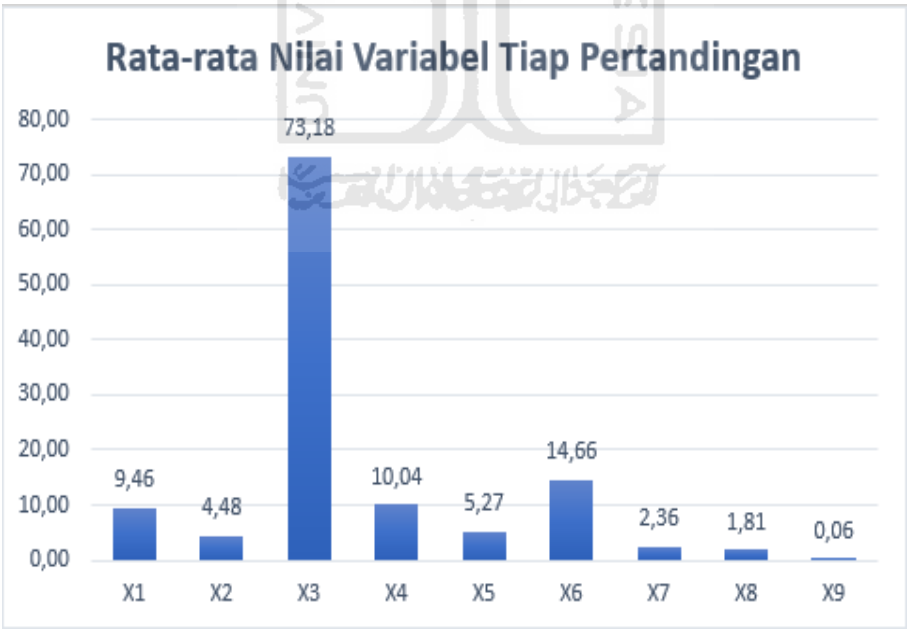

**Gambar 5. 5** Persebaran Rata-rata Umur Pemain Tiap Tim

Nilai pemain adalah harga pemain yang harus ditebus oleh setiap tim jika ingin merekrut suatu pemain ke dalam tim mereka. Nilai pemain biasanya bergantung pada performa pemain di lapangan, umur, posisi, serta prestasi yang sudah dimilikinya. Secara umum semakin tinggi nilai suatu pemain maka semakin bagus pula kualitas yang dimiliki oleh pemain tersebut.




**Gambar 5. 6** Persebaran Total Nilai Pemain Tiap Tim

Pada gambar 5.6 dapat diperhatikan bahwa setiap tim memiliki total nilai pemain yang berbeda-beda dan tim yang memiliki total nilai pemain paling besar pada Liga 1 Indonesia 2018 adalah Bali United FC dengan nilai sebesar Rp. 75,6 milyar sedangkan tim yang memiliki total nilai pemain paling kecil adalah Persebaya Surabaya dengan nilai sebesar Rp. 12 milyar.




**Gambar 5. 7** Rata-rata Nilai Variabel Tiap Pertandingan

Pada gambar 5.7 dapat dilihat bahwa terdapat berbagai variabel yang digunakan dalam analisis klasifikasi pada penelitian ini yaitu total tembakan (x1), tembakan *on target* (x2), akurasi operan (x3), tekel sukses (x4), sepak pojok (x5),

pelanggaran (x6), *offside* (x7), kartu kuning (x8), dan kartu merah (x9). Adapun nilai rata-rata variabel x1 pada setiap pertandingan adalah sebanyak 9,46 total tembakan setiap pertandingan, x2 sebesar 4,48 tembakan *on target* setiap pertandingan, x3 sebesar 73,18% akurasi operan setiap pertandingan, x4 sebesar 10,04 tekel sukses setiap pertandingan, x5 sebesar 5,27 sepak pojok setiap pertandingan, x6 sebesar 14,66 pelanggaran setiap pertandingan, x7 sebesar 2,36 *offside* setiap pertandingan, x8 sebesar 1,81 kartu kuning setiap pertandingan, dan x9 sebesar 0,06 kartu merah setiap pertandingan.

## 4.2 Klasifikasi Menggunakan Data Asli

### 4.2.1 Struktur Data dan Pembagian Data *Training* dan *Testing*

Data yang digunakan pada penelitian kali ini terdiri dari 584 observasi dan 10 variabel yang terbagi dua yaitu variabel dependen (y) dan variabel independen (x). Variabel y bersifat kategorik yang terdiri dari tiga kelas data yaitu hasil menang pada kelas 1, hasil seri pada kelas 2, dan hasil kalah pada kelas 3. Sedangkan variabel x bersifat *integer* dan terbagi menjadi sembilan variabel yaitu total tembakan (x1), tembakan *on target* (x2), akurasi operan (x3), tekel sukses (x4), sepak pojok (x5), pelanggaran (x6), *offside* (x7), kartu kuning (x8), dan kartu merah (x9).

```
> str(data)
'data.frame': 584 obs. of  10 variables:
 $ y : Factor w/ 3 levels "1","2","3": 3 2 1 1 1 1 3 1 1 3 ...
 $ x1: int  7 9 19 11 9 8 8 11 13 8 ...
 $ x2: int  5 4 9 7 2 1 4 4 8 5 ...
 $ x3: int  71 72 80 64 82 67 71 83 79 62 ...
 $ x4: int  11 13 9 16 18 13 7 6 16 11 ...
 $ x5: int  3 7 11 8 6 2 5 6 4 10 ...
 $ x6: int  19 11 15 16 13 15 19 16 19 13 ...
 $ x7: int  1 2 2 3 3 3 2 1 5 4 ...
 $ x8: int  3 1 2 2 2 1 2 3 2 2 ...
 $ x9: int  0 1 0 0 0 0 0 0 0 0 ...
```

**Gambar 5. 8** Tipe Data Variabel Penelitian

Sebelum melakukan klasifikasi, data yang akan digunakan harus dibagi terlebih dahulu ke dalam data *training* dan data *testing*. Pembuatan data *training* sangat diperlukan untuk melatih kinerja metode dalam *machine learning* sedangkan data *testing* digunakan untuk mengukur tingkat akurasi dari model yang sudah

terbentuk di data *training*. Jumlah total dari data yang digunakan pada penelitian ini sebanyak 584 data dengan keterangan kelas 1 terdiri dari 223 data, kelas 2 terdiri dari 138 data, dan kelas 3 terdiri dari 223 data. Peneliti disini melakukan pembagian data dengan perbandingan 70% untuk data *training* dan 30% untuk data *testing*. Pembagian ini dilakukan secara acak dengan menggunakan bantuan *software RStudio* dan didapatkan jumlah data pada data *training* sebanyak 421 data dan data *testing* sebanyak 163 data, untuk melihat hasil pembagiannya dapat diamati pada tabel 5.2.

**Tabel 5. 2** Data *Training* dan Data *Testing* Pada Data Asli

| Keterangan | Kelas 1 (Menang) | Kelas 2 (Seri) | Kelas 3 (Kalah) | Total |
|---|---|---|---|---|
| **Data Training** | 149 | 102 | 170 | 421 |
| **Data Testing** | 74 | 36 | 53 | 163 |
| **Total** | 223 | 138 | 223 | 584 |

### 4.2.2 Metode *Random Forest*

**a. Penentuan *Mtry* dan Jumlah Pohon (*Ntree*) Terbaik**

Sebelum melakukan klasifikasi dengan menggunakan *random forest* terlebih dahulu tentukan nilai parameter yaitu *Mtry* dan *Ntree* yang dibutuhkan dalam membangun model agar didapatkan model terbaik dengan nilai *error* sekecil mungkin. *Mtry* adalah jumlah variabel independen yang digunakan dalam membangun pohon pada setiap iterasi, dalam proses pemilihan nilai *Mtry* terdapat tiga cara yaitu:

1. *Mtry* $= \frac{1}{2}\sqrt{total\ variabel\ independen}$
   $= \frac{1}{2}\sqrt{9} = 3 = 1{,}5 \approx 1$
2. *Mtry* $= \sqrt{total\ variabel\ independen}$
   $= \sqrt{9} = 3$
3. *Mtry* $= 2 \times \sqrt{total\ variabel\ independen}$
   $= 2 \times \sqrt{9} = 6$

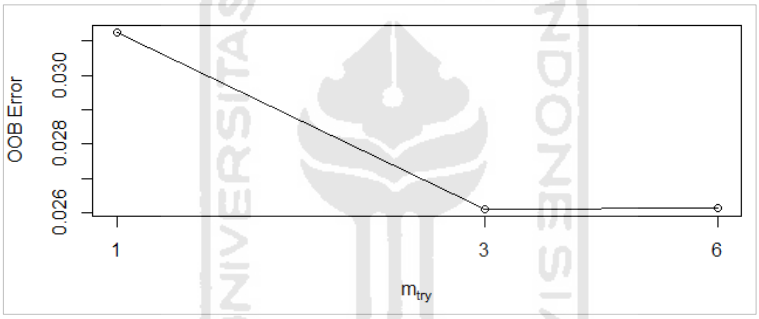
Setelah mendapatkan nilai ketiga *Mtry* tersebut maka lakukan uji coba untuk mencari nilai *error* terkecil dengan melakukan klasifikasi dengan menggunakan

model *default* yang ada pada *software R*. Setelah melakukan percobaan pada setiap nilai Mtry tersebut maka didapatlah *Mtry* terbaik adalah 3 dengan nilai *Error OOB* sebesar 2,6%. Untuk melihat perbandingan nilai *Error OOB* untuk setiap nilai *Mtry* dapat dilihat pada tabel 5.3.

**Tabel 5. 3** Nilai *Error* Tiap *Mtry* Pada Data Asli

| ***Mtry*** | ***OOB Error*** |
|---|---|
| 1 | 3,1239% |
| 3 | 2,6119% |
| 6 | 2,6122% |

Adapun grafik yang dihasilkan dari ketiga nilai *Mtry* tersebut dapat dilihat pada gambar 5.8. Melihat hasil tersebut maka nilai *Mtry* terbaik yang akan digunakan pada model *random forest* yang akan dibangun adalah *Mtry* bernilai 3.




**Gambar 5. 9** Grafik Nilai *Error* Tiap *Mtry* Pada Data Asli

**Tabel 5. 4** Nilai *Error* Tiap *Ntree* Pada Data Asli

| ***Ntree*** | ***OOB Error*** |
|---|---|
| 100 | 15,68% |
| 200 | 16,39% |
| 300 | 15,68% |
| 400 | 15,68% |
| 500 | 16,15% |
| 600 | 16,15% |
| 700 | 15,91% |
| 800 | 16,15% |
| 900 | 16,39% |
| 1000 | 16,15% |

Selanjutnya adalah mencari jumlah pohon (*Ntree*) terbaik yang memiliki nilai *error* terkecil dengan menggunakan nilai *Mtry* terbaik yang sudah didapatkan sebelumnya. Penentuan nilai *Ntree* yang akan diuji tergantung pada peneliti dan

nilai yang akan diuji pada penelitian ini adalah 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000. Adapun nilai *error* pada masing-masing nilai *Ntree* dapat dilihat pada tabel 5.4 yang mana nilai *Ntree* terbaik adalah 400 dengan nilai *error OOB* sebesar 15,68%.

**b. Proses Pelatihan dan Pembentukan Model**

Dari hasil penentuan nilai parameter terbaik sebelumnya yaitu nilai *Mtry* sebesar 3 dan nilai *Ntree* sebesar 400 maka nilai ini akan digunakan pada model *random forest* untuk dilakukan proses klasifikasi.

```
randomForest(formula = y~., data = train, ntree = 400, mtry =3)
                      Type of random forest : classification
                     Number of trees (Ntree): 400
No. of variables tried at each split (Mtry) : 3

        OOB estimate of  error rate: 15.68%
```

**Gambar 5. 10** Hasil Model Terbaik *Random Forest* Pada Data Asli

Pada gambar 5.9 dapat dilihat bahwa tipe *random forest* yang terbentuk adalah klasifikasi dengan jumlah pohon yang dibentuk sebanyak 400 dan jumlah variabel yang digunakan pada setiap iterasinya berjumlah 3 dengan perkiraan tingkat kesalahan *OOB* pada data *training* yang digunakan sebesar 15,68%.

**Tabel 5. 5** Hasil Prediksi Data *Training Random Forest* dengan Data Asli

| **Prediksi** | **Aktual** | | |
|---|---|---|---|
| | **Menang** | **Seri** | **Kalah** |
| **Menang** | 123 | 9 | 15 |
| **Seri** | 6 | 78 | 1 |
| **Kalah** | 20 | 15 | 154 |
| ***Error*** | 17,45% | 23,53% | 9,41% |

Pada tabel 5.5 menyajikan hasil prediksi dari data *training* yang mana tingkat kesalahan prediksi pada hasil menang sebesar 17,45%, hasil seri sebesar 23,53%, dan hasil kalah sebesar 9,41%. Dari ketiga kelas data tersebut kelas yang memiliki kesalahan prediksi paling besar adalah prediksi hasil seri, hal ini disebabkan oleh jumlah data pada kelas hasil seri ini lebih sedikit dibanding kelas yang lain (lihat kembali tabel 5.2) sehingga menyebabkan ketidakseimbangan data (*imbalanced data*) pada setiap kelas yang mempengaruhi tingkat akurasi model dalam memprediksi kelas data.

**c. Pengujian Akurasi Model Dengan Data *Testing***

Setelah model sudah terbentuk pada data *training* maka langkah selanjutnya adalah melakukan pengujian pada data *testing* untuk melihat akurasi dari model yang didapat. Untuk melihat hasil prediksi pada data *testing* lihat tabel 5.6.

**Tabel 5. 6** Hasil Prediksi Data *Testing Random Forest* dengan Data Asli

| **Prediksi** | **Aktual** | | | ***Precision*** |
|---|---|---|---|---|
| | **Menang** | **Seri** | **Kalah** | |
| **Menang** | 68 | 4 | 3 | 90,67% |
| **Seri** | 1 | 27 | 0 | 96,43% |
| **Kalah** | 5 | 5 | 50 | 83,33% |
| ***Sensitivity*** | 91,89% | 75,00% | 94,34% | |

Pada setiap klasifikasi akan menghasilkan tabel prediksi yang mana berupa tabel *confusion matrix* yang menampilkan perbandingan jumlah prediksi dengan data aktual. Dalam mengukur performa hasil prediksi pada *confusion matrix* terdapat berbagai jenis matriks yang bisa dijadikan acuan dalam melihat tingkat kebagusan model dalam memprediksi tiap kelas, diantaranya yaitu *sensitivity* dan *precision*. *Sensitivity* adalah perbandingan antara jumlah prediksi benar dengan total jumlah data aktual pada kelas tertentu, sedangkan *precision* adalah perbandingan antara jumlah prediksi benar dengan total jumlah data prediksi pada kelas tertentu.

Pada tabel 5.6 ditampilkan nilai *sensitivity* dan nilai *precision* tiap kelas yang mana pada nilai *sensitivity* yang memiliki akurasi paling rendah ada pada data kelas hasil seri yang mana nilainya hanya 75%, sedangkan untuk nilai *precision* yang memiliki akurasi paling rendah ada pada prediksi hasil kalah yang mana nilainya sebesar 83,33%. Dari hasil akurasi tersebut dapat diketahui bahwa model yang sudah terbentuk mampu melakukan klasifikasi dengan baik pada kelas data hasil menang dan hasil kalah, namun cenderung buruk dalam melakukan prediksi pada kelas data hasil seri. Adapun rumus untuk mencari total akurasi dari hasil prediksi pada data *testing* adalah sebagai berikut:

$$\text{Total Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{semua prediksi})}$$

$$= \frac{\sum(68+27+50)}{\sum(68+4+3+1+27+0+5+5+50)}$$

$$= \frac{145}{163} = 0{,}8896$$

Secara keseluruhan tingkat akurasi dari model *random forest* yang sudah terbentuk menggunakan data asli untuk melakukan klasifikasi pada hasil prediksi data *testing* adalah sebesar 0,8896 atau 88,96%.

### 4.2.3 Metode *Support Vector Machine*

Dalam melakukan klasifikasi dengan menggunakan metode *support vector machine* (SVM) terdapat beberapa jenis *kernel* yang digunakan dalam mencari model terbaik. Adapun diantaranya yaitu *kernel radial*, *polynomial*, dan *sigmoid*. Pada pembentukkan model setiap *kernel* diperlukan nilai parameter terbaik yang memiliki nilai *error* terkecil. Terdapat dua parameter yang paling berpengaruh secara umum yaitu parameter nilai *cost* dan nilai *gamma*. Nilai *cost* mengatur kompleksitas model dan tingkat kesalahan klasifikasi pada pelatihan model, sedangkan nilai *gamma* mengatur seberapa besar suatu titik data mewakili wilayah yang ada disekitarnya.

Dengan menggunakan jenis *kernel* dan nilai parameter yang tepat maka model yang terbentuk akan menghasilkan akurasi data yang baik. Penentuan nilai *cost* dan nilai *gamma* yang akan diuji tergantung pada peneliti dan nilai yang akan diuji pada penelitian ini untuk nilai *cost* adalah 0,01; 0,05; 0,1; 0,5; 1; 5; 10; 50, sedangkan untuk nilai *gamma* adalah 0,1; 0,5; 1; 1,5; 2. berikut perbandingan hasil akurasi model antara ketiga jenis *kernel* yang diuji:

**a. *Kernel Radial***

**Tabel 5. 7** Nilai *Error* Model Pada *Kernel Radial*

| No | *Cost* | *Gamma* | *Error* | No | *Cost* | *Gamma* | *Error* |
|---|---|---|---|---|---|---|---|
| 1 | 0,01 | 0,1 | 0,5964 | 21 | 1 | 1 | 0,2040 |
| 2 | 0,05 | 0,1 | 0,5846 | 22 | 5 | 1 | 0,2064 |
| 3 | 0,1 | 0,1 | 0,4963 | 23 | 10 | 1 | 0,2064 |
| 4 | 0,5 | 0,1 | 0,4225 | 24 | 50 | 1 | 0,2064 |
| 5 | 1 | 0,1 | 0,4035 | 25 | 0,01 | 1,5 | 0,5964 |
| 6 | 5 | 0,1 | 0,2966 | 26 | 0,05 | 1,5 | 0,5964 |
| 7 | 10 | 0,1 | 0,2635 | 27 | 0,1 | 1,5 | 0,5964 |
| 8 | 50 | 0,1 | 0,1945 | 28 | 0,5 | 1,5 | 0,3445 |
| 9 | 0,01 | 0,5 | 0,5964 | 29 | 1 | 1,5 | 0,2302 |
| 10 | 0,05 | 0,5 | 0,5964 | 30 | 5 | 1,5 | 0,2302 |
| 11 | 0,1 | 0,5 | 0,5964 | 31 | 10 | 1,5 | 0,2302 |
| 12 | 0,5 | 0,5 | 0,3038 | 32 | 50 | 1,5 | 0,2302 |
| 13 | 1 | 0,5 | 0,1802 | 33 | 0,01 | 2 | 0,5964 |
| 14 | 5 | 0,5 | 0,1874 | 34 | 0,05 | 2 | 0,5964 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 15 | 10 | 0,5 | 0,1874 | 35 | 0,1 | 2 | 0,5964 |
| 16 | 50 | 0,5 | 0,1874 | 36 | 0,5 | 2 | 0,3397 |
| 17 | 0,01 | 1 | 0,5964 | 37 | 1 | 2 | 0,2326 |
| 18 | 0,05 | 1 | 0,5964 | 38 | 5 | 2 | 0,2326 |
| 19 | 0,1 | 1 | 0,5964 | 39 | 10 | 2 | 0,2326 |
| 20 | 0,5 | 1 | 0,3348 | 40 | 50 | 2 | 0,2326 |

Berdasarkan tabel 5.7 didapatkan bahwa nilai *cost* sebesar 1 dan nilai *gamma* sebesar 0,5 adalah nilai parameter yang paling optimum untuk model pada *kernel radial* dengan tingkat *error* sebesar 0,1802 atau 18,02%.

```
best.tune(method = svm, train.x = y ~ ., data = train,
ranges = list(cost = c(0.01,0.05,0.1,0.5,1,5,10,50),
gamma  = c(0.1,0.5,1,1.5,2)),kernel = "radial")

Parameters:
  SVM-Type:  C-classification
SVM-Kernel:  radial
      cost:  1
     gamma:  0.5

Number of Support Vectors:  358
```

**Gambar 5. 11** Model Terbaik *Support Vector Machine* Pada *Kernel Radial*

Pada gambar 5.10 ditampilkan model terbaik dari *support vector machine* pada *kernel radial* dengan menggunakan nilai *cost* sebesar 1 dan nilai *gamma* sebesar 0,5. Adapun jumlah *support vector* yang dihasilkan pada model ini untuk membangun *hyperlane* adalah sebanyak 358 *support vectors*.

**Tabel 5. 8** Hasil Prediksi Data *Testing Kernel Radial*

| **Prediksi** | **Aktual** | | | ***Precision*** |
|---|---|---|---|---|
| | **Menang** | **Seri** | **Kalah** | |
| **Menang** | 63 | 6 | 4 | 86,30% |
| **Seri** | 1 | 24 | 0 | 96,00% |
| **Kalah** | 10 | 6 | 49 | 75,38% |
| ***Sensitivity*** | 85,14% | 66,67% | 92,45% | |

Pada tabel 5.8 menampilkan hasil prediksi pada data *testing* dari model SVM pada *kernel radial* yang mana pada nilai *sensitivity* yang memiliki akurasi paling rendah ada pada data kelas hasil seri yang mana nilainya hanya 66,67%, sedangkan untuk nilai *precision* yang memiliki nilai paling rendah ada pada prediksi hasil kalah yang mana nilainya sebesar 75,38%.

Jika dibandingkan dengan hasil prediksi pada metode *random forest* sebelumnya, tingkat kelemahan prediksi model sama yaitu sama-sama memiliki kelemahan dalam memprediksi kelas data hasil seri tapi jika dilihat secara umum pada setiap kelas tingkat akurasinya cenderung lebih rendah dibandingkan dengan metode *random forest*. Adapun rumus untuk mencari total akurasi dari hasil prediksi pada data *testing kernel radial* adalah sebagai berikut:

$$\text{Total Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{semua prediksi})}$$
$$= \frac{\sum(63+24+49)}{\sum(63+6+4+1+24+0+10+6+49)}$$
$$= \frac{136}{163}$$
$$= 0{,}8344$$

Secara keseluruhan tingkat akurasi dari model *support vector machine* pada *kernel radial* yang sudah terbentuk menggunakan data asli untuk melakukan klasifikasi pada hasil prediksi data *testing* adalah sebesar 0,8344 atau 83,44%.

**b. *Kernel Polynomial***

**Tabel 5. 9** Nilai *Error* Model Pada *Kernel Polynomial*

| No | *Cost* | *Gamma* | *Error* | No | *Cost* | *Gamma* | *Error* |
|---|---|---|---|---|---|---|---|
| 1 | 0,01 | 0,1 | 0,5964 | 21 | 1 | 1 | 0,1946 |
| 2 | 0,05 | 0,1 | 0,5678 | 22 | 5 | 1 | 0,1993 |
| 3 | 0,1 | 0,1 | 0,5346 | 23 | 10 | 1 | 0,1993 |
| 4 | 0,5 | 0,1 | 0,4298 | 24 | 50 | 1 | 0,1993 |
| 5 | 1 | 0,1 | 0,4084 | 25 | 0,01 | 1,5 | 0,2348 |
| 6 | 5 | 0,1 | 0,3203 | 26 | 0,05 | 1,5 | 0,2206 |
| 7 | 10 | 0,1 | 0,3014 | 27 | 0,1 | 1,5 | 0,2063 |
| 8 | 50 | 0,1 | 0,2229 | 28 | 0,5 | 1,5 | 0,2017 |
| 9 | 0,01 | 0,5 | 0,3894 | 29 | 1 | 1,5 | 0,1993 |
| 10 | 0,05 | 0,5 | 0,3179 | 30 | 5 | 1,5 | 0,1993 |
| 11 | 0,1 | 0,5 | 0,2800 | 31 | 10 | 1,5 | 0,1993 |
| 12 | 0,5 | 0,5 | 0,2087 | 32 | 50 | 1,5 | 0,1993 |
| 13 | 1 | 0,5 | 0,2087 | 33 | 0,01 | 2 | 0,2158 |
| 14 | 5 | 0,5 | 0,1946 | 34 | 0,05 | 2 | 0,2017 |
| 15 | 10 | 0,5 | 0,1970 | 35 | 0,1 | 2 | 0,1899 |
| 16 | 50 | 0,5 | 0,1993 | 36 | 0,5 | 2 | 0,1993 |
| 17 | 0,01 | 1 | 0,3014 | 37 | 1 | 2 | 0,1993 |
| 18 | 0,05 | 1 | 0,2229 | 38 | 5 | 2 | 0,1993 |
| 19 | 0,1 | 1 | 0,2087 | 39 | 10 | 2 | 0,1993 |
| 20 | 0,5 | 1 | 0,2018 | 40 | 50 | 2 | 0,1993 |

Berdasarkan tabel 5.9 didapatkan bahwa nilai *cost* sebesar 0,1 dan nilai *gamma* sebesar 2 adalah nilai parameter yang paling optimum untuk model pada *kernel polynomial* dengan tingkat *error* sebesar 0,1899 atau 18,99%. Nilai parameter ini kemudian akan digunakan pada model yang akan dibuat selanjutnya untuk dilakukan proses klasifikasi. Pada gambar 5.11 ditampilkan model terbaik dari *support vector machine* pada *kernel polynomial* dengan menggunakan nilai *cost* sebesar 0,1 dan nilai *gamma* sebesar 2. Adapun jumlah *support vector* yang dihasilkan pada model ini untuk membangun *hyperlane* adalah sebanyak 263 *support vectors*.

```
best.tune(method = svm, train.x = y ~ ., data = train,
ranges = list(cost = c(0.01,0.05,0.1,0.5,1,5,10,50),
gamma  = c(0.1,0.5,1,1.5,2)),kernel = "polynomial")

Parameters:
  SVM-Type:  C-classification
SVM-Kernel:  polynomial
      cost:  0.1
     gamma:  2

Number of Support Vectors:  263
```

**Gambar 5. 12** Model Terbaik SVM Pada *Kernel Polynomial*

Setelah didapatkan model optimal pada *kernel polynomial* ini maka selanjutnya diuji dengan menggunakan data *testing* untuk melihat seberapa akurat klasifikasi yang dihasilkan oleh model.

**Tabel 5. 10** Hasil Prediksi Data *Testing Kernel Polynomial*

| **Prediksi** | **Aktual** | | | ***Precision*** |
|---|---|---|---|---|
| | **Menang** | **Seri** | **Kalah** | |
| **Menang** | 61 | 3 | 8 | 84,72% |
| **Seri** | 6 | 27 | 2 | 77,14% |
| **Kalah** | 7 | 6 | 43 | 76,79% |
| ***Sensitivity*** | 82,43% | 75,00% | 81,13% | |

Pada tabel 5.10 menampilkan hasil prediksi pada data *testing* dari model SVM pada *kernel polynomial* yang mana pada nilai *sensitivity* yang memiliki akurasi paling rendah ada pada data kelas hasil seri yang mana nilainya hanya 75%, sedangkan untuk nilai *precision* yang memiliki nilai paling rendah ada pada prediksi hasil kalah yang mana nilainya sebesar 76,79%.

Jika dibandingkan dengan hasil prediksi pada *kernel radial* sebelumnya, tingkat kelemahan prediksi model sama yaitu sama-sama memiliki kelemahan dalam memprediksi kelas data hasil seri sehingga dari hasil analisis sejauh ini yang sudah didapatkan dapat disimpulkan bahwa ketidakseimbangan jumlah data pada setiap kelas akan mempengaruhi model dalam melakukan klasifikasi dengan akurat. Jika dilihat secara umum *kernel polynomial* pada setiap kelas tingkat akurasinya cenderung lebih rendah dibandingkan dengan *kernel radial*. Adapun rumus untuk mencari total akurasi dari hasil prediksi pada data *testing* adalah sebagai berikut:

$$\text{Total Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{semua prediksi})}$$
$$= \frac{\sum(61+27+43)}{\sum(61+3+8+6+27+2+7+6+43)}$$
$$= \frac{131}{163}$$
$$= 0{,}8037$$

Secara keseluruhan tingkat akurasi dari model *support vector machine* pada *kernel polynomial* yang sudah terbentuk menggunakan data asli untuk melakukan klasifikasi pada hasil prediksi data *testing* adalah sebesar 0,8037 atau 80,37%.

**c. *Kernel Sigmoid***

**Tabel 5. 11** Nilai *Error* Model Pada *Kernel Sigmoid*

| No | *Cost* | *Gamma* | *Error* | No | *Cost* | *Gamma* | *Error* |
|---|---|---|---|---|---|---|---|
| 1 | 0,01 | 0,1 | 0,5964 | 21 | 1 | 1 | 0,5697 |
| 2 | 0,05 | 0,1 | 0,4845 | 22 | 5 | 1 | 0,5723 |
| 3 | 0,1 | 0,1 | 0,4631 | 23 | 10 | 1 | 0,5747 |
| 4 | 0,5 | 0,1 | 0,4915 | 24 | 50 | 1 | 0,5748 |
| 5 | 1 | 0,1 | 0,5011 | 25 | 0,01 | 1,5 | 0,5202 |
| 6 | 5 | 0,1 | 0,5650 | 26 | 0,05 | 1,5 | 0,5011 |
| 7 | 10 | 0,1 | 0,5627 | 27 | 0,1 | 1,5 | 0,5484 |
| 8 | 50 | 0,1 | 0,5888 | 28 | 0,5 | 1,5 | 0,5864 |
| 9 | 0,01 | 0,5 | 0,5584 | 29 | 1 | 1,5 | 0,5839 |
| 10 | 0,05 | 0,5 | 0,4798 | 30 | 5 | 1,5 | 0,6056 |
| 11 | 0,1 | 0,5 | 0,5131 | 31 | 10 | 1,5 | 0,5841 |
| 12 | 0,5 | 0,5 | 0,5820 | 32 | 50 | 1,5 | 0,5700 |
| 13 | 1 | 0,5 | 0,5794 | 33 | 0,01 | 2 | 0,5226 |
| 14 | 5 | 0,5 | 0,5939 | 34 | 0,05 | 2 | 0,4868 |
| 15 | 10 | 0,5 | 0,6011 | 35 | 0,1 | 2 | 0,5415 |
| 16 | 50 | 0,5 | 0,5914 | 36 | 0,5 | 2 | 0,5841 |
| 17 | 0,01 | 1 | 0,5346 | 37 | 1 | 2 | 0,6008 |
| 18 | 0,05 | 1 | 0,4726 | 38 | 5 | 2 | 0,5746 |
| 19 | 0,1 | 1 | 0,5342 | 39 | 10 | 2 | 0,5699 |
| 20 | 0,5 | 1 | 0,5747 | 40 | 50 | 2 | 0,5841 |

Berdasarkan tabel 5.11 didapatkan bahwa nilai *cost* sebesar 0,1 dan nilai *gamma* sebesar 0,1 adalah nilai parameter yang paling optimum untuk model pada *kernel sigmoid* dengan tingkat *error* sebesar 0,4631 atau 46,31%. Nilai parameter ini kemudian akan digunakan pada model yang akan dibuat selanjutnya untuk dilakukan proses klasifikasi. Pada gambar 5.12 ditampilkan model terbaik dari *support vector machine* pada *kernel polynomial* dengan menggunakan nilai *cost* sebesar 0,1 dan nilai *gamma* sebesar 0,1. Adapun jumlah *support vector* yang dihasilkan pada model ini untuk membangun *hyperlane* adalah sebanyak 378 *support vectors*.

```
best.tune(method = svm, train.x = y ~ ., data = train,
ranges = list(cost = c(0.01,0.05,0.1,0.5,1,5,10,50),
gamma  = c(0.1,0.5,1,1.5,2)),kernel = "sigmoid")

Parameters:
  SVM-Type:  C-classification
SVM-Kernel:  sigmoid
      cost:  0.1
     gamma:  0.1

Number of Support Vectors:  378
```

**Gambar 5. 13** Model Terbaik *Support Vector Machine* Pada *Kernel Sigmoid*

Setelah didapatkan model optimal pada *kernel sigmoid* ini maka selanjutnya diuji dengan menggunakan data *testing* untuk melihat seberapa akurat klasifikasi yang dihasilkan oleh model.

**Tabel 5. 12** Hasil Prediksi Data *Testing Kernel Sigmoid*

| **Prediksi** | **Aktual** | | | ***Precision*** |
|---|---|---|---|---|
| | **Menang** | **Seri** | **Kalah** | |
| **Menang** | 49 | 14 | 15 | 62,82% |
| **Seri** | 0 | 0 | 0 | 0% |
| **Kalah** | 25 | 22 | 38 | 44,71% |
| ***Sensitivity*** | 66,22% | 0% | 71,70% | |

Pada tabel 5.12 menampilkan hasil prediksi pada data *testing* dari model SVM pada *kernel sigmoid* yang mana pada nilai *sensitivity* yang memiliki akurasi paling rendah ada pada data kelas hasil seri yang mana nilainya 0%, sama halnya untuk nilai *precision* yang memiliki nilai paling rendah ada pada prediksi hasil kalah yang mana nilainya juga 0%. Hal ini menyatakan bahwa *kernel sigmoid* tidak mampu

melakukan klasifikasi pada kelas data hasil seri. Adapun rumus untuk mencari total akurasi dari hasil prediksi pada data *testing kernel sigmoid* adalah sebagai berikut:

$$\text{Total Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{semua prediksi})}$$
$$= \frac{\sum(49+0+38)}{\sum(49+14+15+0+0+0+25+22+38)}$$
$$= \frac{87}{163} = 0{,}5337$$

Secara keseluruhan tingkat akurasi dari model *support vector machine* pada *kernel sigmoid* yang sudah terbentuk menggunakan data asli untuk melakukan klasifikasi pada hasil prediksi data *testing* adalah sebesar 0,5337 atau 53,37%.

## 4.3 Klasifikasi Menggunakan Data *Oversampling*

### 4.3.1 *Balancing Data* dan Pembagian Data *Training* dan *Testing*

Dalam melakukan klasifikasi pada data yang memiliki kelas data yang tidak seimbang (*imbalanced data*) akan menghasilkan klasifikasi dengan akurasi yang rendah sehingga perlu dilakukan penyeimbangan data pada tiap-tiap kelas data yang ada (*balancing data*), salah satu cara melakukan *balancing data* adalah dengan teknik *oversampling*. Teknik *oversampling* adalah metode *sampling* yang melakukan *re-sampling* pada data kelas minoritas dengan cara menambah data kelas minoritas dari data yang sudah ada dengan dipilih secara acak sehingga jumlahnya menjadi sebanyak data kelas mayoritas.

Dari data yang digunakan diketahui bahwa terdapat tiga kelas data yaitu kelas hasil menang berjumlah 223, hasil seri berjumlah 138, dan hasil kalah berjumlah 223. Dikarenakan jumlah kelas data hasil seri tidak memiliki jumlah data yang sama dengan kelas lainnya maka disini akan dilakukan *oversampling* pada kelas data hasil seri sehingga jumlah datanya menjadi sebanyak 223. Setelah dilakukan *oversampling* maka selanjutnya akan dilakukan pembagian data dengan perbandingan 70% untuk data *training* dan 30% untuk data *testing*. Pembagian ini dilakukan secara acak dengan menggunakan bantuan *software RStudio* dan didapatkan jumlah data pada data *training* sebanyak 484 data dan data *testing* sebanyak 185 data, hasil pembagiannya dapat diamati pada tabel 5.13.

**Tabel 5. 13** Data *Training* dan Data *Testing* Pada Data *Oversampling*

| Keterangan | Kelas 1 (Menang) | Kelas 2 (Seri) | Kelas 3 (Kalah) | Total |
|---|---|---|---|---|
| **Data Training** | 162 | 156 | 166 | 484 |
| **Data Testing** | 61 | 67 | 57 | 185 |
| **Total** | 223 | 223 | 223 | 669 |

### 4.3.2 Metode *Random Forest*

**a. Penentuan *Mtry* dan Jumlah Pohon (*Ntree*) Terbaik**

Sebelum melakukan klasifikasi dengan menggunakan *random forest* terlebih dahulu tentukan nilai parameter yaitu *Mtry* dan *Ntree* yang dibutuhkan dalam membangun model agar didapatkan model terbaik dengan nilai *error* sekecil mungkin. *Mtry* adalah jumlah variabel independen yang digunakan dalam membangun pohon pada setiap iterasi, dalam proses pemilihan nilai *Mtry* terdapat tiga cara yaitu:

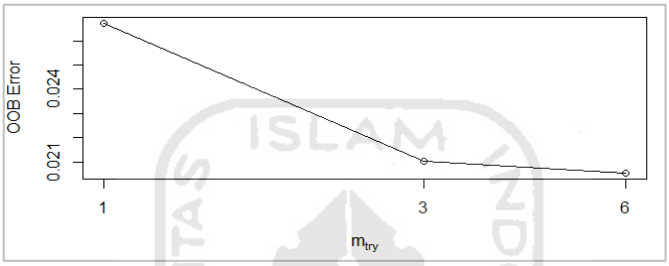
1. *Mtry* $= \frac{1}{2}\sqrt{total\ variabel\ independen}$
   $= \frac{1}{2}\sqrt{9} = 3 = 1{,}5 \approx 1$
2. *Mtry* $= \sqrt{total\ variabel\ independen}$
   $= \sqrt{9} = 3$
3. *Mtry* $= 2 \times \sqrt{total\ variabel\ independen}$
   $= 2 \times \sqrt{9} = 6$

Setelah mendapatkan nilai ketiga *Mtry* tersebut maka lakukan uji coba untuk mencari nilai *error* terkecil dengan melakukan klasifikasi dengan menggunakan model *default* yang ada pada *software R*. Setelah melakukan percobaan pada setiap nilai Mtry tersebut maka didapatlah *Mtry* terbaik adalah 6 dengan nilai *Error OOB* sebesar 2,6%. Untuk melihat perbandingan nilai *Error OOB* untuk setiap nilai *Mtry* dapat dilihat pada tabel 5.14.

**Tabel 5. 14** Nilai *Error* Tiap *Mtry* Pada Data *Oversampling*

| ***Mtry*** | **OOB *Error*** |
|---|---|
| 1 | 2,67% |
| 3 | 2,10% |
| 6 | 2,06% |

Adapun grafik yang dihasilkan dari ketiga nilai *Mtry* tersebut dapat dilihat pada gambar 5.13. Melihat hasil tersebut maka nilai *Mtry* terbaik yang akan digunakan pada model *random forest* yang akan dibangun adalah *Mtry* bernilai 6.




**Gambar 5. 14** Grafik Nilai *Error* Tiap *Mtry* Pada Data *Oversampling*

Selanjutnya adalah mencari jumlah pohon (*Ntree*) terbaik yang memiliki nilai *error* terkecil dengan menggunakan nilai *Mtry* terbaik yang sudah didapatkan sebelumnya. Penentuan nilai *Ntree* yang akan diuji tergantung pada peneliti dan nilai yang akan diuji pada penelitian ini adalah 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000. Adapun nilai *error* pada masing-masing nilai *Ntree* dapat dilihat pada tabel 5.15 yang mana nilai *Ntree* terbaik adalah 1000 dengan nilai *error OOB* sebesar 10,95%.

**Tabel 5. 15** Nilai *Error* Tiap *Ntree* Pada Data *Oversampling*

| ***Ntree*** | **OOB *Error*** |
|---|---|
| 100 | 11,36% |
| 200 | 11,98% |
| 300 | 11,78% |
| 400 | 11,36% |
| 500 | 11,36% |
| 600 | 11,78% |
| 700 | 11,36% |
| 800 | 11,16% |
| 900 | 11,16% |
| 1000 | 10,95% |

**b. Proses Pelatihan dan Pembentukan Model**

Dari hasil penentuan nilai parameter terbaik sebelumnya yaitu nilai *Mtry* sebesar 6 dan nilai *Ntree* sebesar 1000 maka nilai ini akan digunakan pada model *random forest* untuk dilakukan proses klasifikasi.

```
randomForest(formula= y~., data= train, ntree= 1000, mtry=6)
                      Type of random forest : classification
                    Number of trees (Ntree): 1000
No. of variables tried at each split (Mtry) : 6

        OOB estimate of  error rate: 10.95%
```

**Gambar 5. 15** Hasil Model Terbaik *Random Forest* Pada Data *Oversampling*

Pada gambar 5.14 dapat dilihat bahwa tipe *random forest* yang terbentuk adalah klasifikasi dengan jumlah pohon yang dibentuk sebanyak 1000 dan jumlah variabel yang digunakan pada setiap iterasinya berjumlah 6 dengan perkiraan tingkat kesalahan *OOB* pada data *training* yang digunakan sebesar 10,95%.

**Tabel 5. 16** Hasil Prediksi Data *Training* dengan Data *Oversampling*

| **Prediksi** | **Aktual** | | |
|---|---|---|---|
| | **Menang** | **Seri** | **Kalah** |
| **Menang** | 135 | 2 | 11 |
| **Seri** | 10 | 149 | 8 |
| **Kalah** | 17 | 5 | 147 |
| ***Error*** | 16,67% | 4,49% | 11,45% |

Pada tabel 5.16 menyajikan hasil prediksi dari data *training* yang mana tingkat kesalahan prediksi pada hasil menang sebesar 16,67%, hasil seri sebesar 4,49%, dan hasil kalah sebesar 11,45%. Dari tingkat *error* tiga kelas data tersebut jika dibandingkan dengan hasil data asli maka hasil data *oversampling* secara keseluruhan memiliki tingkat *error* yang lebih rendah sehingga proses *balancing data* yang dilakukan sebelumnya terbukti memberi pengaruh terhadap tingkat akurasi model dalam melakukan klasifikasi terhadap data yang digunakan.

**c. Pengujian Akurasi Model Dengan Data *Testing***

Setelah model sudah terbentuk pada data *training* maka langkah selanjutnya adalah melakukan pengujian pada data *testing* untuk melihat akurasi dari model yang didapat. Untuk melihat hasil prediksi pada data *testing* lihat tabel 5.17.

**Tabel 5. 17** Hasil Prediksi Data *Testing* dengan Data *Oversampling*

| **Prediksi** | **Aktual** | | | ***Precision*** |
|---|---|---|---|---|
| | **Menang** | **Seri** | **Kalah** | |
| **Menang** | 53 | 3 | 2 | 91,38% |
| **Seri** | 3 | 64 | 3 | 91,43% |
| **Kalah** | 5 | 0 | 52 | 91,23% |
| ***Sensitivity*** | 86,89% | 95,52% | 91,23% | |

Pada setiap klasifikasi akan menghasilkan tabel prediksi yang mana berupa tabel *confusion matrix* yang menampilkan perbandingan jumlah prediksi dengan data aktual. Dalam mengukur performa hasil prediksi pada *confusion matrix* terdapat berbagai jenis matriks yang bisa dijadikan acuan dalam melihat tingkat kebagusan model dalam memprediksi tiap kelas, diantaranya yaitu *sensitivity* dan *precision*. *Sensitivity* adalah perbandingan antara jumlah prediksi benar dengan total jumlah data aktual pada kelas tertentu, sedangkan *precision* adalah perbandingan antara jumlah prediksi benar dengan total jumlah data prediksi pada kelas tertentu.

Pada tabel 5.17 ditampilkan nilai *sensitivity* dan nilai *precision* tiap kelas yang mana secara keseluruhan dapat diketahui bahwa model yang terbentuk sudah mampu melakukan klasifikasi dengan baik pada setiap kelas yang ada. Adapun rumus untuk mencari total akurasi dari hasil prediksi pada data *testing* adalah sebagai berikut:

$$\text{Total Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{semua prediksi})}$$
$$= \frac{\sum(53+64+52)}{\sum(53+3+2+3+64+3+5+0+52)}$$
$$= \frac{169}{185}$$
$$= 0{,}9135$$

Secara keseluruhan tingkat akurasi dari model *random forest* yang sudah terbentuk menggunakan data *oversampling* untuk melakukan klasifikasi pada hasil prediksi data *testing* adalah sebesar 0,9135 atau 91,35%.

### 4.3.3 Metode *Support Vector Machine*

Dalam melakukan klasifikasi dengan menggunakan metode *support vector machine* (SVM) terdapat beberapa jenis *kernel* yang digunakan dalam mencari

model terbaik. Adapun diantaranya yaitu *kernel radial*, *polynomial*, dan *sigmoid*. Pada pembentukkan model setiap *kernel* diperlukan nilai parameter terbaik yang memiliki nilai *error* terkecil. Terdapat dua parameter yang paling berpengaruh secara umum yaitu parameter nilai *cost* dan nilai *gamma*. Nilai *cost* mengatur kompleksitas model dan tingkat kesalahan klasifikasi pada pelatihan model, sedangkan nilai *gamma* mengatur seberapa besar suatu titik data mewakili wilayah yang ada disekitarnya.

Dengan menggunakan jenis *kernel* dan nilai parameter yang tepat maka model yang terbentuk akan menghasilkan akurasi data yang baik. Penentuan nilai *cost* dan nilai *gamma* yang akan diuji tergantung pada peneliti dan nilai yang akan diuji pada penelitian ini untuk nilai *cost* adalah 0,01; 0,05; 0,1; 0,5; 1; 5; 10; 50, sedangkan untuk nilai *gamma* adalah 0,1; 0,5; 1; 1,5; 2. berikut perbandingan hasil akurasi model antara ketiga jenis *kernel* yang diuji:

**a. *Kernel Radial***

**Tabel 5. 18** Nilai *Error* Model Pada *Kernel Radial*

| No | Cost | Gamma | Error | No | Cost | Gamma | Error |
|---|---|---|---|---|---|---|---|
| 1 | 0,01 | 0,1 | 0,6759 | 21 | 1 | 1 | 0,1595 |
| 2 | 0,05 | 0,1 | 0,6201 | 22 | 5 | 1 | 0,1615 |
| 3 | 0,1 | 0,1 | 0,5165 | 23 | 10 | 1 | 0,1615 |
| 4 | 0,5 | 0,1 | 0,4342 | 24 | 50 | 1 | 0,1615 |
| 5 | 1 | 0,1 | 0,3968 | 25 | 0,01 | 1,5 | 0,6636 |
| 6 | 5 | 0,1 | 0,3143 | 26 | 0,05 | 1,5 | 0,6636 |
| 7 | 10 | 0,1 | 0,2813 | 27 | 0,1 | 1,5 | 0,6636 |
| 8 | 50 | 0,1 | 0,1987 | 28 | 0,5 | 1,5 | 0,1783 |
| 9 | 0,01 | 0,5 | 0,6677 | 29 | 1 | 1,5 | 0,1657 |
| 10 | 0,05 | 0,5 | 0,6677 | 30 | 5 | 1,5 | 0,1636 |
| 11 | 0,1 | 0,5 | 0,6677 | 31 | 10 | 1,5 | 0,1636 |
| 12 | 0,5 | 0,5 | 0,2317 | 32 | 50 | 1,5 | 0,1636 |
| 13 | 1 | 0,5 | 0,1884 | 33 | 0,01 | 2 | 0,6636 |
| 14 | 5 | 0,5 | 0,1718 | 34 | 0,05 | 2 | 0,6636 |
| 15 | 10 | 0,5 | 0,1676 | 35 | 0,1 | 2 | 0,6636 |
| 16 | 50 | 0,5 | 0,1676 | 36 | 0,5 | 2 | 0,1782 |
| 17 | 0,01 | 1 | 0,6616 | 37 | 1 | 2 | 0,1699 |
| 18 | 0,05 | 1 | 0,6616 | 38 | 5 | 2 | 0,1658 |
| 19 | 0,1 | 1 | 0,6616 | 39 | 10 | 2 | 0,1658 |
| 20 | 0,5 | 1 | 0,1676 | 40 | 50 | 2 | 0,1658 |

Berdasarkan tabel 5.18 didapatkan bahwa nilai *cost* sebesar 1 dan nilai *gamma* sebesar 1 adalah nilai parameter yang paling optimum untuk model pada *kernel radial* dengan tingkat *error* sebesar 0,1595 atau 15,95%.

```
best.tune(method = svm, train.x = y ~ ., data = train,
ranges = list(cost = c(0.01,0.05,0.1,0.5,1,5,10,50),
gamma  = c(0.1,0.5,1,1.5,2)),kernel = "radial")

Parameters:
  SVM-Type:  C-classification
SVM-Kernel:  radial
      cost:  1
     gamma:  1

Number of Support Vectors:  373
```

**Gambar 5. 16** Model Terbaik *Support Vector Machine* Pada *Kernel Radial*

Pada gambar 5.15 ditampilkan model terbaik dari *support vector machine* pada *kernel radial* dengan menggunakan nilai *cost* sebesar 1 dan nilai *gamma* sebesar 1. Adapun jumlah *support vector* yang dihasilkan pada model ini untuk membangun *hyperlane* adalah sebanyak 373 *support vectors*.

**Tabel 5. 19** Hasil Prediksi Data *Testing Kernel Radial*

| **Prediksi** | **Aktual** | | | ***Precision*** |
|---|---|---|---|---|
| | **Menang** | **Seri** | **Kalah** | |
| **Menang** | 56 | 1 | 7 | 87,50% |
| **Seri** | 0 | 60 | 0 | 100,00% |
| **Kalah** | 5 | 6 | 50 | 81,97% |
| ***Sensitivity*** | 91,80% | 89,55% | 87,72% | |

Pada tabel 5.19 menampilkan hasil prediksi pada data *testing* dari model SVM pada *kernel radial* yang mana pada nilai *sensitivity* dan nilai *precision* tiap kelas secara keseluruhan dapat diketahui bahwa model yang terbentuk sudah mampu melakukan klasifikasi dengan baik pada setiap kelas yang ada. Jika dibandingkan dengan hasil prediksi pada metode *random forest* sebelumnya, tingkat akurasi prediksi model pada *kernel* ini sudah sama-sama memiliki akurasi yang baik. Adapun rumus untuk mencari total akurasi dari hasil prediksi pada data *testing kernel radial* adalah sebagai berikut:

$$\text{Total Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{semua prediksi})}$$

$$= \frac{\sum(56+60+50)}{\sum(56+1+7+0+60+0+5+6+50)}$$

$$= \frac{166}{185}$$

$$= 0{,}8973$$

Secara keseluruhan tingkat akurasi dari model *support vector machine* pada *kernel radial* yang sudah terbentuk menggunakan data *oversampling* untuk melakukan klasifikasi pada hasil prediksi data *testing* adalah sebesar 0,8973 atau 89,73%.

**b. *Kernel Polynomial***

Selanjutnya jenis kernel yang kedua yaitu *kernel polynomial* yang mana berdasarkan pada tabel 5.20 didapatkan bahwa nilai *cost* sebesar 1 dan nilai *gamma* sebesar 1 adalah nilai parameter yang paling optimum untuk model pada *kernel polynomial* ini yang mana dengan nilai tersebut didapatkan besaran nilai *error* sebesar 0,1697 atau 16,97%. Jika dibandingkan dengan nilai *error* pada *kernel radial* sebelumnya yang memiliki nilai error 15,95% maka nilai *error* pada *kernel polynomial* ini sedikit lebih tinggi. Nilai parameter ini kemudian akan digunakan pada model yang akan dibuat selanjutnya untuk dilakukan proses klasifikasi.

**Tabel 5. 20** Nilai *Error* Model Pada *Kernel Polynomial*

| No | *Cost* | *Gamma* | *Error* | No | *Cost* | *Gamma* | *Error* |
|---|---|---|---|---|---|---|---|
| 1 | 0,01 | 0,1 | 0,6881 | 21 | 1 | 1 | 0,1697 |
| 2 | 0,05 | 0,1 | 0,5889 | 22 | 5 | 1 | 0,1739 |
| 3 | 0,1 | 0,1 | 0,5455 | 23 | 10 | 1 | 0,1739 |
| 4 | 0,5 | 0,1 | 0,4484 | 24 | 50 | 1 | 0,1739 |
| 5 | 1 | 0,1 | 0,4029 | 25 | 0,01 | 1,5 | 0,2086 |
| 6 | 5 | 0,1 | 0,3121 | 26 | 0,05 | 1,5 | 0,1780 |
| 7 | 10 | 0,1 | 0,2750 | 27 | 0,1 | 1,5 | 0,1841 |
| 8 | 50 | 0,1 | 0,2045 | 28 | 0,5 | 1,5 | 0,1718 |
| 9 | 0,01 | 0,5 | 0,3864 | 29 | 1 | 1,5 | 0,1739 |
| 10 | 0,05 | 0,5 | 0,2977 | 30 | 5 | 1,5 | 0,1739 |
| 11 | 0,1 | 0,5 | 0,2583 | 31 | 10 | 1,5 | 0,1739 |
| 12 | 0,5 | 0,5 | 0,1984 | 32 | 50 | 1,5 | 0,1739 |
| 13 | 1 | 0,5 | 0,1841 | 33 | 0,01 | 2 | 0,1984 |
| 14 | 5 | 0,5 | 0,1758 | 34 | 0,05 | 2 | 0,1821 |
| 15 | 10 | 0,5 | 0,1739 | 35 | 0,1 | 2 | 0,1738 |
| 16 | 50 | 0,5 | 0,1739 | 36 | 0,5 | 2 | 0,1739 |
| 17 | 0,01 | 1 | 0,2750 | 37 | 1 | 2 | 0,1739 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 18 | 0,05 | 1 | 0,2045 | 38 | 5 | 2 | 0,1739 |
| 19 | 0,1 | 1 | 0,1923 | 39 | 10 | 2 | 0,1739 |
| 20 | 0,5 | 1 | 0,1862 | 40 | 50 | 2 | 0,1739 |

Selanjutnya jika diamati pada gambar 5.16 didapatkan bahwa dengan menggunakan nilai parameter *cost* dengan nilai 1 dan parameter *gamma* dengan nilai 1 yang merupakan nilai yang paling optimum untuk model pada *kernel polynomial* ternyata *support vector* yang dihasilkan pada model ini untuk membangun *hyperlane* adalah sebanyak 300 *support vectors* yang mana jumlah ini lebih sedikit jika dibandingkan pada *kernel radial*.

```
best.tune(method = svm, train.x = y ~ ., data = train,
ranges = list(cost = c(0.01,0.05,0.1,0.5,1,5,10,50),
gamma  = c(0.1,0.5,1,1.5,2)),kernel = "polynomial")

Parameters:
  SVM-Type:  C-classification
SVM-Kernel:  polynomial
      cost:  1
     gamma:  1

Number of Support Vectors:  300
```

**Gambar 5. 17** Model Terbaik SVM Pada *Kernel Polynomial*

Setelah didapatkan model optimal pada *kernel polynomial* ini maka selanjutnya diuji dengan menggunakan data *testing* untuk melihat seberapa akurat klasifikasi yang dihasilkan oleh model.

**Tabel 5. 21** Hasil Prediksi Data *Testing Kernel Polynomial*

| **Prediksi** | **Aktual** | | | ***Precision*** |
|---|---|---|---|---|
| | **Menang** | **Seri** | **Kalah** | |
| **Menang** | 50 | 1 | 6 | 87,72% |
| **Seri** | 2 | 64 | 5 | 90,14% |
| **Kalah** | 9 | 2 | 46 | 80,70% |
| ***Sensitivity*** | 81,97% | 95,52% | 80,70% | |

Pada tabel 5.21 menampilkan hasil prediksi pada data *testing* dari model SVM pada *kernel polynomial* yang mana pada nilai *sensitivity* yang memiliki akurasi paling rendah ada pada data kelas hasil kalah yang mana nilainya 80,70%,

sedangkan untuk nilai *precision* yang memiliki akurasi paling rendah ada pada prediksi hasil kalah yang mana nilainya juga sebesar 80,70%.

Jika dibandingkan dengan hasil prediksi pada *kernel radial* sebelumnya, tingkat prediksi model pada *kernel polynomial* cenderung lebih buruk atau nilainya lebih rendah. Adapun rumus untuk mencari total akurasi dari hasil prediksi pada data *testing* adalah sebagai berikut:

$$\text{Total Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{semua prediksi})}$$

$$= \frac{\sum(50+64+46)}{\sum(50+1+6+2+64+5+9+2+46)}$$

$$= \frac{160}{185}$$

$$= 0{,}8649$$

Secara keseluruhan tingkat akurasi dari model *support vector machine* pada *kernel polynomial* yang sudah terbentuk menggunakan data *oversampling* untuk melakukan klasifikasi pada hasil prediksi data *testing* adalah sebesar 0,8649 atau 86,49%. Melihat nilai ini dibandingkan pada *kernel* sebelumnya yaitu *kernel radial* maka penggunaan *kernel polynomial* lebih buruk dalam melakukan klasifikasi terhadap data *oversampling*.

**c. *Kernel Sigmoid***

**Tabel 5. 22** Nilai *Error* Model Pada *Kernel Sigmoid*

| No | *Cost* | *Gamma* | *Error* | No | *Cost* | *Gamma* | *Error* |
|---|---|---|---|---|---|---|---|
| 1 | 0,01 | 0,1 | 0,6739 | 21 | 1 | 1 | 0,6278 |
| 2 | 0,05 | 0,1 | 0,5413 | 22 | 5 | 1 | 0,6094 |
| 3 | 0,1 | 0,1 | 0,4917 | 23 | 10 | 1 | 0,6114 |
| 4 | 0,5 | 0,1 | 0,5207 | 24 | 50 | 1 | 0,6237 |
| 5 | 1 | 0,1 | 0,5496 | 25 | 0,01 | 1,5 | 0,5517 |
| 6 | 5 | 0,1 | 0,5830 | 26 | 0,05 | 1,5 | 0,5432 |
| 7 | 10 | 0,1 | 0,5764 | 27 | 0,1 | 1,5 | 0,5702 |
| 8 | 50 | 0,1 | 0,5827 | 28 | 0,5 | 1,5 | 0,6317 |
| 9 | 0,01 | 0,5 | 0,5848 | 29 | 1 | 1,5 | 0,6257 |
| 10 | 0,05 | 0,5 | 0,5125 | 30 | 5 | 1,5 | 0,6029 |
| 11 | 0,1 | 0,5 | 0,5599 | 31 | 10 | 1,5 | 0,6155 |
| 12 | 0,5 | 0,5 | 0,6199 | 32 | 50 | 1,5 | 0,6154 |
| 13 | 1 | 0,5 | 0,5969 | 33 | 0,01 | 2 | 0,5537 |
| 14 | 5 | 0,5 | 0,5700 | 34 | 0,05 | 2 | 0,5517 |
| 15 | 10 | 0,5 | 0,5722 | 35 | 0,1 | 2 | 0,5804 |
| 16 | 50 | 0,5 | 0,5971 | 36 | 0,5 | 2 | 0,6069 |
| 17 | 0,01 | 1 | 0,5538 | 37 | 1 | 2 | 0,6050 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 18 | 0,05 | 1 | 0,5329 | 38 | 5 | 2 | 0,6110 |
| 19 | 0,1 | 1 | 0,5784 | 39 | 10 | 2 | 0,5944 |
| 20 | 0,5 | 1 | 0,6236 | 40 | 50 | 2 | 0,6006 |

Berdasarkan tabel 5.22 didapatkan bahwa nilai *cost* sebesar 0,1 dan nilai *gamma* sebesar 0,1 adalah nilai parameter yang paling optimum untuk model pada *kernel sigmoid* dengan tingkat *error* sebesar 0,4917 atau 49,17%. Nilai parameter ini kemudian akan digunakan pada model yang akan dibuat selanjutnya untuk dilakukan proses klasifikasi. Pada gambar 5.17 ditampilkan model terbaik dari *support vector machine* pada *kernel sigmoid* dengan menggunakan nilai *cost* sebesar 0,1 dan nilai *gamma* sebesar 0,1. Dengan *support vector* sebanyak 468.

```
best.tune(method = svm, train.x = y ~ ., data = train,
ranges = list(cost = c(0.01,0.05,0.1,0.5,1,5,10,50),
gamma  = c(0.1,0.5,1,1.5,2)),kernel = "sigmoid")

Parameters:
  SVM-Type:  C-classification
SVM-Kernel:  sigmoid
      cost:  0.1
     gamma:  0.1

Number of Support Vectors:  468
```

**Gambar 5. 18** Model Terbaik *Support Vector Machine* Pada *Kernel Sigmoid*

Setelah didapatkan model optimal pada *kernel sigmoid* ini maka selanjutnya diuji dengan menggunakan data *testing* untuk melihat seberapa akurat klasifikasi yang dihasilkan oleh model.

**Tabel 5. 23** Hasil Prediksi Data *Testing Kernel Sigmoid*

| **Prediksi** | **Aktual** | | | ***Precision*** |
|---|---|---|---|---|
| | **Menang** | **Seri** | **Kalah** | |
| **Menang** | 40 | 19 | 5 | 62,50% |
| **Seri** | 7 | 16 | 16 | 41,03% |
| **Kalah** | 14 | 32 | 36 | 43,90% |
| ***Sensitivity*** | 65,57% | 23,88% | 63,16% | |

Pada tabel 5.23 menampilkan hasil prediksi pada data *testing* dari model SVM pada *kernel sigmoid* yang mana pada nilai *sensitivity* yang memiliki akurasi paling rendah ada pada data kelas hasil seri yang mana nilainya 23,88%, dan juga untuk nilai *precision* yang memiliki akurasi paling rendah ada pada prediksi hasil seri

yang mana nilainya 41,03%. Jika dibandingkan dengan menggunakan data asli maka metode SVM dengan *kernel sigmoid* ini jauh lebih bagus karena meski nilai akurasinya masih rendah tapi model ini masih mampu melakukan klasifikasi tidak seperti pada data asli yang sama-sama juga menggunakan *kernel sigmoid* tapi gagal mengklasifikasikan kelas data hasil seri. Adapun rumus untuk mencari total akurasi dari hasil prediksi pada data *testing kernel sigmoid* adalah sebagai berikut:

$$\text{Total Akurasi} = \frac{\sum(\text{prediksi benar})}{\sum(\text{semua prediksi})}$$
$$= \frac{\sum(40+16+36)}{\sum(40+19+5+7+16+16+14+32+36)}$$
$$= \frac{92}{185} = 0{,}4973$$

Secara keseluruhan tingkat akurasi dari model *support vector machine* pada *kernel sigmoid* yang sudah terbentuk menggunakan data *oversampling* untuk melakukan klasifikasi pada hasil prediksi data *testing* adalah sebesar 49,73%.

## 4.4 Perbandingan Semua Hasil Setiap Metode Klasifikasi

Selanjutnya untuk melihat hasil setiap metode yang telah dilakukan dan mencari metode yang memiliki akurasi model tertinggi maka dapat dilihat perbandingan dari semuanya pada tabel 5.24.

**Tabel 5. 24** Perbandingan Akurasi Model Antar Metode Klasifikasi

| **Metode** | **Data Asli** | | | |
|---|---|---|---|---|
| ***Randon Forest*** | **Hasil Pertandingan** | ***Sensitivity*** | ***Precision*** | **Total Akurasi** |
| | Menang | 91,89% | 90,67% | **88,96%** |
| | Seri | 75% | 96,43% | |
| | Kalah | 94,34% | 83,33% | |
| ***Support Vector Machine*** | **Hasil Pertandingan** | ***Sensitivity*** | ***Precision*** | **Total Akurasi** |
| ***Radial*** | Menang | 85,14% | 86,30% | **83,44%** |
| | Seri | 66,67% | 96% | |
| | Kalah | 92,45% | 75,38% | |
| ***Polynomial*** | Menang | 82,43% | 84,72% | **80,37%** |
| | Seri | 75% | 77,14% | |
| | Kalah | 81,13% | 76,79% | |
| ***Sigmoid*** | Menang | 66,22% | 62,82% | **53,37%** |

| | | | | |
|---|---|---|---|---|
| | Seri | 0% | 0% | |
| | Kalah | 71,70% | 44,71% | |
| **Metode** | **Data *Oversampling*** | | | |
| ***Randon Forest*** | **Hasil Pertandingan** | ***Sensitivity*** | ***Precision*** | **Total Akurasi** |
| | Menang | 86,89% | 91,38% | **91,35%** |
| | Seri | 95,52% | 91,43% | |
| | Kalah | 91,23% | 91,23% | |
| ***Support Vector Machine*** | **Hasil Pertandingan** | ***Sensitivity*** | ***Precision*** | **Total Akurasi** |
| ***Radial*** | Menang | 91,80% | 87,50% | **89,73%** |
| | Seri | 89,55% | 100% | |
| | Kalah | 87,72% | 81,97% | |
| ***Polynomial*** | Menang | 81,97% | 87,72% | **86,49%** |
| | Seri | 95,52% | 90,14% | |
| | Kalah | 80,70% | 80.70% | |
| ***Sigmoid*** | Menang | 65,57% | 62,50% | **49,73%** |
| | Seri | 23,88% | 41,03% | |
| | Kalah | 63,16% | 43,90% | |

Berdasarkan tabel 5.24 metode yang memiliki tingkat total akurasi model terbesar adalah metode *random forest* dengan menggunakan data *oversampling* yaitu dengan akurasi model sebesar 91,35% unggul terhadap metode *support vector machine* (SVM) dengan akurasi model sebesar 89,73% yang sama-sama menggunakan data *oversampling*. Sehingga dalam tahap analisis selanjutnya yang akan peneliti pakai adalah metode random forest dengan menggunakan data *oversampling*, dan juga dalam hal ini mempertegas bahwa keseimbangan antar kelas data sangat mempengaruhi akurasi dari model yang akan dibentuk sehingga perlu adanya proses *balancing* data yang mana hal ini sudah dilakukan pada data *oversampling*.

## 4.5 Ukuran Tingkat Variabel Terpenting

Pada tahap analisis selanjutnya dilakukan pengurutan variabel independen yang terpenting (*importance variable*) yang mana dapat dilihat pada tabel 5.25.

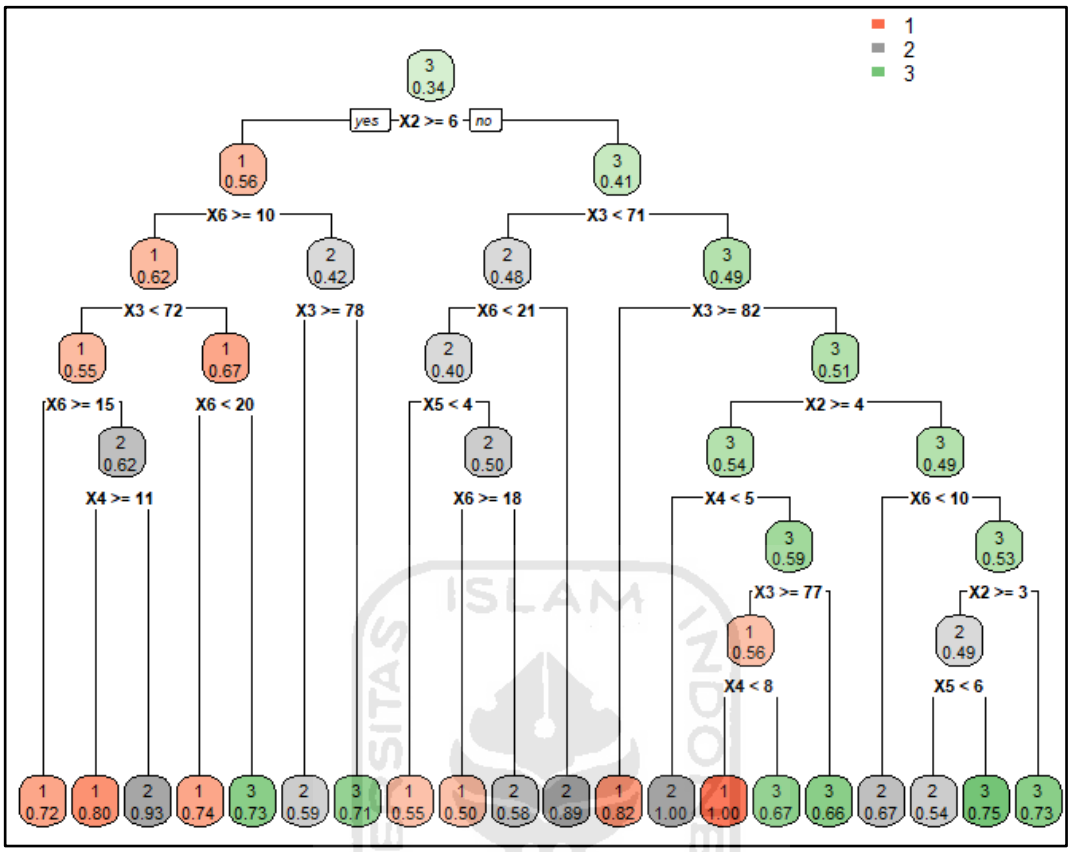
**Tabel 5. 25** Variabel Terpenting Dengan *Random Forest*

| Variabel Independen (x) | *Mean Decrease Accuracy* |
|---|---|
| Tembakan *On Target* (x2) | 128,26 |
| Akurasi Operan (x3) | 114,55 |
| Tekel Sukses (x4) | 83,88 |
| Sepak Pojok (x5) | 81,33 |
| Pelanggaran (x6) | 81,30 |
| Total Tembakan (x1) | 69,87 |
| *Offside* (x7) | 67,33 |
| Kartu Kuning (x8) | 56,34 |
| Kartu Merah (x9) | 17,82 |

Variabel terpenting (*importance variable*) dari metode pengklasifikasian dengan *random forest* menunjukkan tingkat kepentingan suatu variabel didalam model klasifikasi yang sudah dibuat. Salah satu ukuran dalam penentuan tingkat kepentingan adalah *mean decrease accuracy* (MDA) yang mana menampilkan seberapa besar tambahan observasi yang mengalami kesalahan klasifikasi jika salah satu variabel independen tidak diikutsertakan ke dalam pengujian. Semakin tinggi nilai MDA dari suatu variabel independen tersebut maka semakin penting pula pengaruhnya dalam akurasi model klasifikasi yang dibentuk.

Pada tabel 5.25 variabel independen yang terpenting dalam model klasifikasi *random forest* yang dibuat adalah tembakan *on target* (x2), lalu diikuti oleh akurasi operan (x3), tekel sukses (x4), sepak pojok (x5), pelanggaran (x6), total tembakan (x1), *offside* (x7), kartu kuning (x8), dan ditutup oleh kartu merah (x9) yang mana kesemuanya itu diurutkan dari nilai yang terbesar ke yang terkecil.

Adapun salah satu hasil *decision tree* (pohon keputusan) dari model dengan menggunakan metode *random forest* data *oversampling* yang sudah dihasilkan dapat dilihat pada gambar 5.19.




**Gambar 5. 19** Hasil *Decision Tree* Dari Model *Random Forest*

Diagram yang berwarna merah muda melambang data kelas 1 (hasil menang), warna abu-abu melambangkan data kelas 2 (hasil seri), dan warna hijau melambangkan data kelas 3 (hasil kalah). Sebagai salah satu contoh, berdasarkan gambar 5.18 suatu tim akan memenangkan pertandingan jika memiliki syarat pertama yaitu nilai x2 (tembakan *on target*) lebih besar sama dengan 6 lalu syarat kedua jika nilai x6 (pelanggaran) lebih besar sama dengan 10 lalu syarat ketiga jika nilai x3 (akurasi operan) kurang dari 72 ataupun lebih dari 72 lalu syarat keempat jika nilai x6 (pelanggaran) kurang dari 20 maka tim tersebut memiliki peluang memenangkan pertandingan sebesar 74%. Sama halnya untuk setiap kasus keadaaan hasil menang lain, hasil seri, dan hasil kalah sesuai yang terdapat pada gambar 5.19 diatas.

Terdapat hal unik dari gambar 5.19 diatas yang mana ada suatu kondisi dimana suatu tim akan memenangkan pertandingan jika memiliki jumlah pelanggaran lebih dari 10, hal ini dapat menjadi intropeksi bagi para wasit di liga 1

Indonesia untuk lebih bijaksana dalam mengambil keputusan dilapangan, sebab jika hal ini dibiarkan maka liga Indonesia akan menjadi liga yang penuh pelanggaran dari setiap klub-klub yang bermain untuk meraih kemenangan. Dari segi para pemain maupun klub pun juga harus peduli terhadap hal ini, sebab dengan adanya banyak pelanggaran maka resiko cedera yang akan dialami para pemain juga akan semakin tinggi dan bahkan beresiko karir mereka akan hancur karena cedera tersebut, oleh karena itu tentu perlu diperbaiki lagi kedepannya dengan kerja sama yang baik dengan berbagai pihak.

# BAB VI
# KESIMPULAN DAN SARAN

## 5.1 Kesimpulan

Dari hasil analisis yang telah dilakukan terhadap data hasil pertandingan sepak bola Liga 1 Indonesia musim kompetisi 2018 dapat diperoleh beberapa kesimpulan sebagai berikut:

1. Faktor-faktor yang mempengaruhi kemenangan suatu tim dalam pertandingan sepak bola adalah total tembakan, tembakan *on target*, akurasi operan, tekel sukses, sepak pojok, pelanggaran, *offside*, kartu kuning, dan kartu merah. Selain itu ada pula faktor non teknis yang juga ikut berpengaruh seperti klub yang bermain dikandang atau tandang dan apakah pelatih yang digunakan diklub pelatih lokal atau asing yang mana dari hasil penelitian diketahui bahwa klub memikiki tingkat kemenangan yang lebih tinggi jika bermain di kandang serta klub yang menggunakan pelatih asing.
2. Metode yang paling tepat dalam mengklasifikasi data yang digunakan adalah metode *random forest* dengan menggunakan *data oversampling* dengan menggunakan *Mtry* sebanyak 6 dan *Ntree* sebanyak 1000 dengan tingkat akurasi sebesar 91,35%.
3. Variabel yang paling berpengaruh dalam model klasifikasi *random forest* yang dibuat adalah tembakan *on target* (x2), lalu diikuti oleh akurasi operan (x3), tekel sukses (x4), sepak pojok (x5), pelanggaran (x6), total tembakan (x1), *offside* (x7), kartu kuning (x8), dan ditutup oleh kartu merah (x9).

## 5.2 Saran

1. Dalam kesempatan berikutnya penulis mau memberi saran pada pihak selanjutnya yang ingin melakukan penelitian terhadap sepakbola liga Indonesia untuk dapat lebih meningkatkan ataupun menambah penggunaan berbagai variabel yang mungkin turut serta mempengaruhi tingkat kemenangan suatu tim dalam memenangkan pertandingan karena keterbatasan data yang ada,

penulis disini hanya mencantumkan 9 variabel dalam proses analisis ini dan penulis harap kedepannya agar bisa ditambah lagi sehingga hasil analisis akan jauh lebih bagus dan hasilnya pun akan lebih kaya akan informasi guna kepentingan suatu tim dalam menyusun strategi dalam permainan.

2. Pihak Liga 1 Indonesia diharapkan kedepannya agar dapat meningkatkan *database* pertandingan agar data yang disajikan pada setiap pertandingan menjadi lebih banyak lagi sehingga informasi yang dihasilkan akan lebih baik lagi.
3. Untuk penelitian selanjutnya mungkin para peneliti dapat menfokuskan pada timnas Indonesia agar kualitas timnas menjadi lebih baik dan meraih banyak trofi kedepannya.
4. Dari penelitian ini dapat kita ketahui bahwa membeli pemain bintang belum tentu akan membuat suatu menjadi lebih kuat dalam memenangkan permainan sebab masih banyak faktor yang perlu dipertimbangkan sehingga dalam kebijakan transfer pemain klub diharapkan membuat keputusan yang lebih matang lagi.

# DAFTAR PUSTAKA


AFC. 2020. *AFC Club Competitions Ranking.* http://www.the-afc.com/afc-ranking/. Diakses tanggal 16 Mei 2020, pukul 16.11 WIB.

Angga, Ferry. 2013. *Aplikasi Statistik Dalam Sepak Bola*. http://fni-statistics.blogspot.com/2013/06/aplikasi-statistik-dalam-sepak-bola.html. Diakses tanggal 14 Oktober 2020, pukul 15.35 WIB.

Attenberg, J. dan Ertekin, S. 2013. *Class Imbalanced and Active Learning*. *In H. He & Y, Ma. Imbalanced Learning : Foundations, Algorithms, and Applications,* New Jersey : John Wiley & Sons.

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

Bramer, M. 2013. *Principles of Data Mining Second Edition*. London: Springer.

Breiman, L. 2000. *Statistical modeling: The two cultures (with comments and a rejoinder by the author)*. Statistical Science, 16(3):199–231.

Breiman, L. 2001. *Random Forest. Machine Learning* 45, 5-32.

Breiman, L. dan Cutler, A. 2003. *Manual on Setting Up, Using, and Understanding Random Forest V4.0*. http://oz.berkeley.edu/users/breiman/Using_random_forests_v4.0.pdf. Diakses tanggal 23 Mei 2020, pukul 17.21 WIB.

Chandra, Andreas. 2017. *Perbedaan Supervised and Unsupervised Learning*. https://datascience.or.id/article/Perbedaan-Supervised-and-Unsupervised-Learning-5a8fa6e6. Diakses tanggal 17 Mei 2020, pukul 20.39 WIB.

Dewi, Nariswari Karina. 2011. *Penerapan Metode Random Forest dalam Driver Analysis*. Bogor: Institut Pertanian Bogor.

Gorunescu, F. 2011. *Data Mining: Concepts, Models, and Techniques*. New York: Springer-Verlag.

Han, J dan Kamber, M. 2006. *Data Mining Concepts and Techniques Second Edition*. San Fransisco: Diane Cerra.

Harjono, Yulvianus. 2018. *Menambang Talenta Lewat Statistik dan Analisis Data*. https://ligakg.kompas.id/baca/2018/03/01/menambang-talenta-lewat-

statistik-dan-analisis-data. Diakses tanggal 14 oktober 2020, pukul 15.51 WIB.

Hastie, T., Tibshirani, R. dan Friedman, J. 2008. *The Elements of Statistics Learning: Data Mining, Inference, dan Prediction*. California: Springer.

Jain, Kunal. 2015. *Machine Learning Basics for a Newbie*. https://www.analyticsvidhya.com/blog/2015/06/machine-learning-basics/. Diakses tanggal 17 Mei 2020, pukul 20.44 WIB.

Japkowicz, N. 2000. *The Class Imbalance Problem: Significance and Strategies*. In Proceedings of the 200 International Conference on Artificial Intelligence (IC-AI 2000): Special Track on Inductive Learning Las Vegas, Nevada.

Jatmiko, Y. A., Padmadisastra, S. dan Chadidjah, A. 2017. *Perbandingan Teknik Sampling Dalam Random Forest Pada Kelas Imbalanced*. Seminar Statistika FMIPA UNPAD 2017 (SNS VI). ISSN : 2087-2590.

Koehrsen, Will. 2017. *Random Forest Simple Explanation*. https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d. Diakses tanggal 19 Mei 2020, pukul 18.18 WIB.

Laia, Yonata., Tandian, Charles dan Saputra, Andi. 2019. *Penerapan Data Mining dalam Memprediksi Pemenang Klub Sepak Bola Pada Ajang Liga Champion dengan Algoritma C.45.* Medan: Universitas Prima Indonesia.

Liaw, A. dan Wiener, M. 2002. *Classification and Regression by Random Forest*. R News, 2, 18-22.

Liu, X.Y. dan Zhou, Z.H. 2013. *Ensemble Methods for Class Imbalance Learning In H. He & Y, Ma. Imbalanced Learning : Foundations, Algorithms, and Applications,* New Jersey: John Wiley & Sons.

Mambang dan Byna, A. 2017. *Analisis Perbandingan Algoritma C.45, Random Forest dengan CHILD Decision Tree Untuk Klasifikasi Tingkat Kecemasan Ibu Hamil*. Seminar Nasional Teknologi Informasi dan Multimedia 2017.

Nugroho, A.S., Witarto, A.B. dan Handoko, D. 2003. *Support Vector Machine: Teori dan Aplikasinya dalam Bioinformatika*. Proceeding of Indonesia Scientific Meeting in Central Japan.

Octaviani, Puspita Anna., Wilandari, Yuciana dan Ispriyanti, Dwi. 2014. *Penerapan Metode Klasifikasi Support Vector Machine (SVM) Pada Data Akreditasi Sekolah Dasar (SD) Di Kabupaten Magelang.* Semarang: Universitas Diponegoro.

Prasetio, R.T. dan Pratiwi. 2015. *Penerapan Teknik Bagging Pada Algoritma Klasifikasi Untuk Mengatasi Ketidakseimbangan Kelas Dataset Medis*. Jurnal Informatika. Vol.II No.2 Hal.395-403.

Pratiwi, Y. R. 2017. *Perbandingan Analisis Sentimen Pada Petralite Melalui Jejaring Sosial Twitter dengan Menggunakan Metode Support Vector Machine dan Maximum Entropy*. Yogyakarta: Universitas Islam Indonesia.

Putra, Eka Permana. 2017. *Belajar Mengenal Istilah Statistik Dalam Sepakbola*. https://sleman-football.com/belajar-mengenal-istilah-statistik-dalam-sepakbola-bagian-1/. Diakses tanggal 14 Oktober 2020, pukul 15.08 WIB.

Putra, Erzha Aulia. 2017. *Evaluasi Faktor-Faktor Yang Mempengaruhi Kemenangan Dalam Pertandingan Sepak Bola Dengan Menggunakan Pohon Klasifikasi.* Bogor: Institut Pertanian Bogor.

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

Sastrawan, A. S., Baizal, Z. A. dan Bijaksana, M. A. (2010). *Analisis Pengaruh Metode Combine Sampling Dalam Churn Prediction Untuk Perusahaan Telekomunikasi*. Seminar Nasional Informatika 2010 UPN Veteran Yogyakarta.

Sembiring, K. 2007. *Penerapan Teknik Support Vector Machine Untuk Pendeteksian Intrusi Pada Jaringan*. Bandung: Institut Teknologi Bandung.

Walpole, R.E. dan Myers, R.H. 1995. *Ilmu Peluang dan Statistika untuk Insinyur dan Ilmuan Edisi ke-4.* Bandung: Institut Teknologi Bandung.

Wang, S. dan Yao, X. 2013. *Using Class Imbalance Learning for Software Defect Prediction.* IEEE Transactions on Reliability, 434-443.

Wezel, M. V. dan Pothaarst. 2007. *European Journal of Operational Research 2007*. Vol.181. Issue 1, Hal. 436-452.

Yasinaron. 2017. *Makalah Statistik dalam Penjaskes*. https://yasinaron1545.blogspot.com/2017/01/makalah-statistik-dalam-penjaskes.html. Diakses tanggal 16 Mei 2020, pukul 18.22 WIB.

Zhang, H. dan Wang, Z. 2011. *A Normal Distribution Based Over-Sampling Approach to Imbalanced Data Classification Advanced Data Mining and Application*. 7th Internation Conference. Beijing: Springer.

# DAFTAR LAMPIRAN

**Lampiran 1** Data Penelitian

| No. | Y | X1 | X2 | X3 | X4 | X5 | X6 | X7 | X8 | X9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 7 | 5 | 71 | 11 | 3 | 19 | 1 | 3 | 0 |
| 2 | 2 | 9 | 4 | 72 | 13 | 7 | 11 | 2 | 1 | 1 |
| 3 | 1 | 19 | 9 | 80 | 9 | 11 | 15 | 2 | 2 | 0 |
| 4 | 1 | 11 | 7 | 64 | 16 | 8 | 16 | 3 | 2 | 0 |
| 5 | 1 | 9 | 2 | 82 | 18 | 6 | 13 | 3 | 2 | 0 |
| 6 | 1 | 8 | 1 | 67 | 13 | 2 | 15 | 3 | 1 | 0 |
| 7 | 3 | 8 | 4 | 71 | 7 | 5 | 19 | 2 | 2 | 0 |
| 8 | 1 | 11 | 4 | 83 | 6 | 6 | 16 | 1 | 3 | 0 |
| 9 | 1 | 13 | 8 | 79 | 16 | 4 | 19 | 5 | 2 | 0 |
| 10 | 3 | 8 | 5 | 62 | 11 | 10 | 13 | 4 | 2 | 0 |
| 11 | 1 | 12 | 7 | 70 | 14 | 8 | 12 | 1 | 3 | 0 |
| 12 | 1 | 19 | 7 | 79 | 8 | 4 | 13 | 1 | 2 | 0 |
| 13 | 3 | 1 | 1 | 72 | 15 | 2 | 15 | 4 | 3 | 0 |
| 14 | 3 | 10 | 5 | 66 | 3 | 13 | 19 | 6 | 4 | 0 |
| 15 | 1 | 14 | 6 | 65 | 18 | 9 | 14 | 2 | 0 | 0 |
| 16 | 1 | 12 | 5 | 78 | 5 | 8 | 20 | 2 | 2 | 1 |
| 17 | 1 | 15 | 8 | 78 | 11 | 6 | 12 | 1 | 3 | 0 |
| 18 | 3 | 11 | 6 | 84 | 9 | 8 | 15 | 3 | 2 | 0 |
| 19 | 3 | 12 | 6 | 72 | 7 | 5 | 23 | 0 | 2 | 0 |
| 20 | 2 | 7 | 1 | 69 | 2 | 6 | 7 | 1 | 1 | 0 |
| 21 | 3 | 12 | 3 | 75 | 9 | 10 | 8 | 3 | 0 | 0 |
| 22 | 3 | 4 | 3 | 69 | 10 | 4 | 11 | 0 | 0 | 0 |
| 23 | 3 | 5 | 3 | 64 | 11 | 3 | 19 | 1 | 1 | 0 |
| 24 | 1 | 7 | 5 | 68 | 10 | 3 | 18 | 3 | 3 | 0 |
| 25 | 3 | 4 | 1 | 82 | 14 | 3 | 7 | 2 | 0 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 561 | 3 | 11 | 5 | 77 | 15 | 6 | 18 | 0 | 2 | 0 |
| 562 | 1 | 11 | 4 | 78 | 11 | 10 | 14 | 0 | 2 | 0 |
| 563 | 2 | 9 | 6 | 72 | 6 | 10 | 12 | 3 | 1 | 1 |
| 564 | 3 | 9 | 5 | 75 | 11 | 6 | 17 | 1 | 3 | 0 |
| 565 | 1 | 6 | 4 | 77 | 7 | 2 | 17 | 10 | 1 | 0 |
| 566 | 3 | 11 | 7 | 78 | 8 | 8 | 20 | 0 | 0 | 0 |
| 567 | 3 | 10 | 9 | 80 | 4 | 9 | 15 | 0 | 2 | 0 |
| 568 | 3 | 4 | 2 | 70 | 9 | 1 | 16 | 3 | 3 | 0 |
| 569 | 2 | 14 | 7 | 80 | 14 | 3 | 15 | 1 | 5 | 0 |
| 570 | 3 | 6 | 3 | 77 | 6 | 11 | 8 | 4 | 1 | 0 |
| 571 | 3 | 12 | 7 | 78 | 12 | 8 | 8 | 0 | 2 | 0 |
| 572 | 2 | 6 | 3 | 70 | 13 | 4 | 9 | 8 | 3 | 0 |
| 573 | 3 | 7 | 4 | 75 | 15 | 5 | 12 | 0 | 0 | 0 |
| 574 | 1 | 15 | 7 | 82 | 6 | 2 | 24 | 1 | 2 | 0 |
| 575 | 1 | 4 | 1 | 70 | 4 | 2 | 14 | 1 | 3 | 0 |
| 576 | 1 | 15 | 9 | 68 | 14 | 10 | 9 | 3 | 1 | 0 |
| 577 | 1 | 8 | 3 | 73 | 4 | 1 | 12 | 2 | 2 | 0 |
| 578 | 1 | 9 | 5 | 68 | 11 | 5 | 20 | 4 | 3 | 0 |
| 579 | 1 | 9 | 5 | 68 | 11 | 5 | 20 | 4 | 3 | 0 |
| 580 | 1 | 6 | 2 | 79 | 10 | 8 | 14 | 0 | 0 | 0 |
| 581 | 2 | 10 | 5 | 77 | 3 | 6 | 11 | 5 | 0 | 0 |
| 582 | 2 | 6 | 3 | 75 | 12 | 2 | 15 | 4 | 1 | 0 |
| 583 | 3 | 9 | 5 | 75 | 11 | 6 | 17 | 1 | 3 | 0 |
| 584 | 1 | 11 | 6 | 76 | 10 | 3 | 16 | 3 | 3 | 1 |

Keterangan Y (Hasil Pertandingan):
Menang = 1
Seri = 2
Kalah = 3

X1 = Total Tembakan
X2 = Tembakan On Target
X3 = Akurasi Operan
X4 = Tekel Sukses
X5 = Sepak Pojok
X6 = Pelanggaran
X7 = Offiside
X8 = Kartu Kuning
X9 = Kartu Merah

**Lampiran 2** *Syntax Random Forest*

```
# Read Data
data <- read.delim("clipboard")
str(data)
data$Y <- as.factor(data$Y)
str(data)
table(data$Y)

# Data Partition
set.seed(1304)
ind <- sample(2, nrow(data), replace = TRUE, prob = c(0.7, 0.3))
train <- data[ind==1,]
test <- data[ind==2,]

# Random Forest
library(randomForest)
set.seed(1304)
rf <- randomForest(Y~., data=train,
                   ntree = 1000,
                   mtry = 6,
                   importance = TRUE,
                   proximity = TRUE)

print(rf)

# Error rate of Random Forest
plot(rf)
```

**Lampiran 3** *Syntax Mtry* terbaik, akurasi data *training* dan *testing*

```
# Tune mtry
set.seed(1304)
t <- tuneRF(train[,-10], train[,10],
            stepFactor = 0.5,
            plot = TRUE,
            ntreeTry = 1000,
            trace = TRUE,
            improve = 0.05)

# Prediction & Confusion Matrix - train data
library(caret)
p1 <- predict(rf, train)
confusionMatrix(p1, train$Y)

# Prediction & Confusion Matrix - test data
p2 <- predict(rf, test)
confusionMatrix(p2, test$Y)

# Variable Importance
varImpPlot(rf)
importance(rf)
```

**Lampiran 4** *Syntax Decision Tree*

```
# Read Data
dataku <- read.delim("clipboard")
str(dataku)
dataku$Y <- as.factor(dataku$Y)
str(dataku)
table(dataku$Y)

# Data Partition
set.seed(1304)
ind <- sample(2, nrow(dataku), replace = TRUE, prob = c(0.70, 0.30))
train <- dataku[ind==1,]
test <- dataku[ind==2,]

#Decision Tree with rpart
library(rpart)
mytree <- rpart(Y~X2+X3+X4+X5+X6,train)
library(rpart.plot)
rpart.plot(mytree, extra=8)
```

**Lampiran 5** *Syntax Support Vector Machine*

```
# Read Data
mydata <- read.delim("clipboard")
str(mydata)
mydata$Y <- as.factor(mydata$Y)
str(mydata)
table(mydata$Y)

# Data Partition
set.seed(1304)
ind <- sample(2, nrow(mydata), replace = TRUE, prob = c(0.70, 0.30
train <- mydata[ind==1,]
test <- mydata[ind==2,]

#Support Vector Machine
library(e1071)
```

**Lampiran 6** *Syntax Kernel Radial*

```
#Tuning
#Kernel radial
set.seed(1304)
tmodel1 <- tune(svm, Y~., data=train, kernel = "radial",
           ranges = list(cost = c(0.01,0.05,0.1,0.5,1,5,10,50),
                         gamma= c(0.1,0.5,1,1.5,2)))
summary(tmodel1)
mymodel1 <- tmodel1$best.model
summary(mymodel1)

#Confusion Matrix and Accuracy
pred <- predict(mymodel1, test)
tab <- table(Predicted = pred, Actual = test$Y)
tab
sum(diag(tab))/sum(tab)
```

**Lampiran 7** *Syntax Kernel Polynomial* dan *Sigmoid*

```
#Kernel polynomial
set.seed(1304)
tmodel3 <- tune(svm, Y~., data=train, kernel = "polynomial",
           ranges = list(cost = c(0.01,0.05,0.1,0.5,1,5,10,50),
                         gamma= c(0.1,0.5,1,1.5,2)))
summary(tmodel3)
mymodel3 <- tmodel3$best.model
summary(mymodel3)

#Confusion Matrix and Accuracy
pred <- predict(mymodel3, test)
tab <- table(Predicted = pred, Actual = test$Y)
tab
sum(diag(tab))/sum(tab)

#Kernel sigmoid
set.seed(1304)
tmodel4 <- tune(svm, Y~., data=train, kernel = "sigmoid",
           ranges = list(cost = c(0.01,0.05,0.1,0.5,1,5,10,50),
                         gamma= c(0.1,0.5,1,1.5,2)))
summary(tmodel4)
mymodel4 <- tmodel4$best.model
summary(mymodel4)

#Confusion Matrix and Accuracy
pred <- predict(mymodel4, test)
tab <- table(Predicted = pred, Actual = test$Y)
tab
sum(diag(tab))/sum(tab)
```