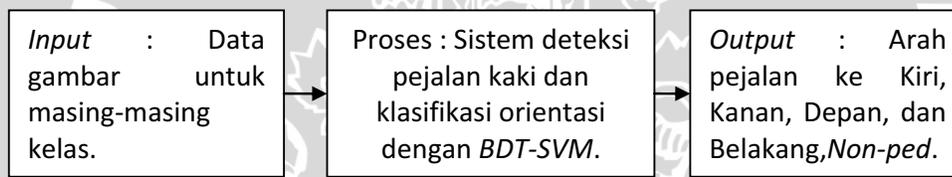


BAB 2 LANDASAN KEPUSTAKAAN

Pada bab ini berisi dasar-dasar teori yang akan dibahas dan digunakan untuk menunjang topik-topik pembahasan. Dasar teori memberikan informasi mengenai beberapa teori yang dibutuhkan untuk penyusunan skripsi ini.

2.1 Kajian Pustaka

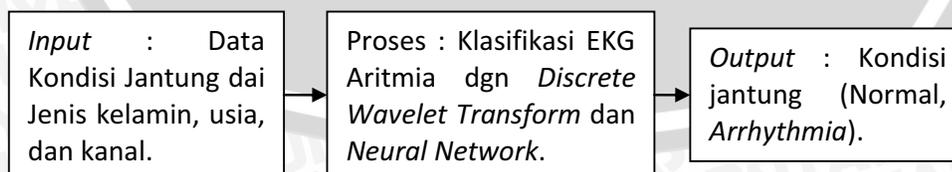
Pada kajian pustaka ini membahas mengenai beberapa penelitian sebelumnya yang dijadikan referensi untuk melakukan penelitian. Pada penelitian sebelumnya yaitu "*Detect Pedestrian Orientation by Integrating Multiclass SVM Utilizing Binary Decision Tree*" (G. Santoshi & G. Gowri P., 2015). Penelitian ini untuk memprediksi kemungkinan terjadi tabrakan dengan mendeteksi arah pejalan kaki menggunakan metode yang sama yaitu *binary decision tree-SVM* menggunakan dataset *INRIA* yang terdiri dari 2000 gambar. Hasil akurasi untuk *Non-Ped* (Bukan pejalan kaki) sebesar 100%, depan dan kiri sebesar 99%, kanan sebesar 95%, dan belakang sekitar 90%. Diagram blok dari penelitian ini bisa dilihat pada Gambar 2.1 di bawah ini.



Gambar 2.1 Diagram Blok *Detect Pedestrian Orientation by Integrating Multiclass SVM Utilizing Binary Decision Tree*.

Sumber : G. Santoshi & G. Gowri P. (2015).

Selain penelitian di atas, penelitian dengan judul "*Classification of ECG Arrhythmias using Discrete Wavelet Transform and Neural Networks*" (Maedeh Kiani Sarkaleh & Asadollah Shahbahrami, 2012). Pada penelitian ini memiliki topik penelitian yang sama yaitu klasifikasi kondisi jantung dari pemeriksaan EKG tetapi dengan metode yang berbeda yaitu menggunakan metode JST dan *Discrete Wavelet Transforms* yang terdiri dari 500 data. Hasilnya memiliki tingkat akurasi 96,5%, ketika 13 *neuron* berada pada lapisan tersembunyi di *traingdx*, 11 *neuron* dalam algoritma *trainrp* dan 14 *neuron* di *trainlm*. Diagram blok dari penelitian ini bisa dilihat pada Gambar 2.2 di bawah ini.



Gambar 2.2 Diagram Blok *Classification of ECG Arrhythmias using Discrete Wavelet Transform and Neural Networks*.

Sumber : Maedeh Kiani Sarkaleh & Asadollah Shahbahrami (2012).

Untuk lebih memahami metode maupun objek yang peneliti gunakan, maka peneliti melakukan kajian terhadap beberapa penelitian lainnya yang berhubungan dengan metode maupun objek yang peneliti angkat. Salah satu kelebihan dari metode SVM adalah mampu menangani kasus dengan *input space* yang berdimensi tinggi (Gjorgji Madzarov, Dejan Gjorgjevikj, dkk, 2008). Metode *Binary Decision Tree* ditambahkan untuk mengatasi kelemahan dari SVM yang hanya *binary class*. Pada Tabel 2.1 Tabel Review Kajian Pustaka terdapat juga penelitian yang dilakukan oleh beberapa peneliti dengan berbagai judul, seperti di bawah ini.

Tabel 2.1 Tabel Review Kajian Pustaka.

No.	Judul	Objek (<i>Input</i>)	Metode (<i>Proses</i>)	Hasil (<i>Output</i>)
1	<i>A Multi-class SVM Classifier Utilizing Binary Decision Tree</i>	Parameter <i>input</i> yaitu : 1. <i>MNIST</i> (Pengenalan digit tulisan tangan terisolasi) 2. <i>Pendigit (UCI Repository)</i> 3. <i>Optdigit (UCI Repository)</i> 4. <i>Statlog</i> (Pengenalan surat dengan tulisan tangan)	Dalam menyelesaikan permasalahan klasifikasi <i>binary</i> , menggunakan metode SVM dengan <i>Gaussian Kernel</i> . Selain SVM juga menggunakan <i>Binary Decision Tree</i> Selain menggunakan pendekatan di atas, penelitian ini juga menggunakan metode <i>neural network</i> .	Metode SVM menghasilkan solusi yang lebih cepat dan lebih baik dibandingkan dengan metode <i>non-SVM</i> . Kelas nya yaitu σ dan C. Nilai akurasi untuk metode <i>OVO</i> dan <i>DAGSVM</i> mencapai tingkat kesalahan sekitar 1,5% lebih tinggi dibandingkan dengan metode <i>OVA</i> .
2	<i>Content-Based Image Retrieval Using a Quick SVM-Binary Decision Tree – QSVMBDT</i>	Parameter <i>input</i> yaitu : 1. warna, 2. tekstur, 3. tepi garis dan 4. bentuk Dari parameter diatas diekstrak dari 1000 gambar. Gambar ini terdiri dari Kategori semantik yang berbeda yaitu gambar wajah	1. Untuk proses klasifikasi menggunakan metode <i>SVM-BDT</i> 2. Menggunakan metode <i>Fuzzy C-Mean (FCM)</i> dan <i>SVM-PWC</i> untuk <i>prefiltering</i> gambar <i>DB</i> . metode di atas merupakan metode dalam arsitektur <i>QSVMBDT</i> .	Penelitian ini mengusulkan sebuah kerangka Cepat <i>SVM BDT</i> , yang melakukan suatu pengambilan efisien dan cepat menggunakan biner arsitektur pohon keputusan berdasarkan SVM dan <i>prefiltering</i> . Nilai akurasi metode <i>SVM-PWC</i> sebesar 74,5%, akurasi <i>SVM-BDT</i>

		Afrika, pantai laut pemandangan, bangunan, bus, dinosaurus, gajah, mawar, kuda, pegunungan dan piring.		sebesar 97,6%, dan akurasi <i>QSVMBDT</i> sebesar 97,6%.
3	<i>Evaluation of Distance Measures for Multi-class Classification in Binary SVM Decision Tree</i>	Parameter <i>input</i> yaitu : <ol style="list-style-type: none"> 1. <i>MNIST</i> (Pengenalan digit tulisan tangan terisolasi) 2. <i>Pendigit</i> (UCI Repository) 3. <i>Optdigit</i> (UCI Repository). 4. Jumlah data yaitu 6000. 	Metode yang digunakan adalah <i>SVM-BDT</i> . <i>BDT</i> digunakan untuk mengatasi permasalahan <i>SVM</i> yang hanya bisa menangani dua kelas. Jarak menggunakan <i>Euclidian Distance</i> , <i>Standardized Euclidean distance</i> dan <i>Mahalanobis distance</i> .	Akurasi prediksi sebagai kriteria kinerja pengklasifikasi menunjukkan bahwa <i>Mahalanobis Distance</i> adalah ukuran jarak yang paling cocok untuk mengukur kesamaan antara kelas dalam proses pengelompokan membangun arsitektur <i>classifier</i> membandingkan dengan langkah-langkah jarak lain dari metode <i>SVM-BDT</i> .
4	Klasifikasi Kondisi Jantung menggunakan JST Berdasarkan Pemodelan Sinyal <i>Electrocardiography</i> .	Parameter <i>input</i> yaitu dari data pemeriksaan Hasil <i>ECG</i> . Data diambil dari <i>MIT BIH</i> Aritmia sejumlah 48 data yang dibedakan berdasarkan jenis kelamin, usia, dan jenis kanal yang terbagi menjadi dua yaitu <i>MLII</i> dan <i>VI</i> .	Proses perhitungan dengan menggunakan Jaringan Syaraf Tiruan. <ol style="list-style-type: none"> 1. Proses pemilihan orde model. 2. Setelah itu melakukan klasifikasi dengan Jaringan Syaraf Tiruan. 	<i>Output</i> : Kondisi fisiologis dari Normal, <i>Ventricular Tachycardia</i> , <i>Ventricular Bigeminy</i> , dan <i>Atrial Fibrillation</i> . Klasifikasi untuk data uji tidak sama dengan data latih sebesar 66,67% untuk kondisi <i>Atrial Fibrillation</i> , 75% untuk kondisi <i>Ventricular Bigeminy</i> , 40% untuk kondisi <i>Ventricular Tachycardia</i> , dan

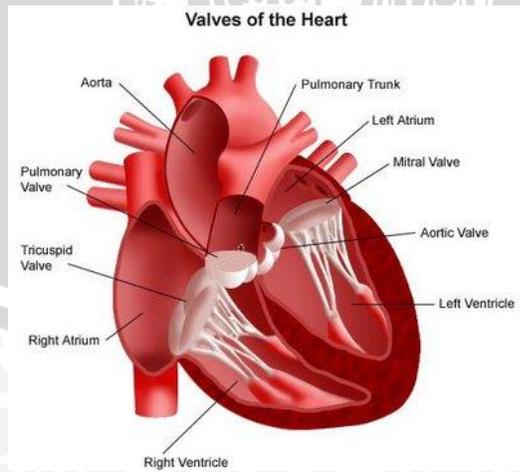
				30% untuk kondisi normal.
5	Klasifikasi Kondisi Detak Jantung Berdasarkan Hasil Pemeriksaan Elektrokardiografi (EKG) menggunakan Binary Decision Tree - Support Vector Machine (BDT-SVM).	Parameter <i>input</i> yaitu: 1. Nilai sadapan <i>MLII</i> dan <i>VI</i> Data diambil dari <i>MIT-BIH</i> Aritmia dengan jumlah data 140.	Metode yang digunakan adalah <i>SVM-BDT</i> . <i>BDT</i> digunakan untuk mengatasi permasalahan <i>SVM</i> yang hanya bisa menangani dua kelas.	Keluaran dari sistem ini yaitu hasil klasifikasi dari kelas normal, <i>atrial fibrillation</i> , <i>pvc bigeminy</i> , dan <i>ventricular tachycardia</i> .

Sumber : Gjorgji Madzarov *et all* (2008), Felci Rajam & S. Valli (2011), Gjorgji Madzarov & Dejan Gjorgjevikj, dan Edwin, dkk (2011).

2.2 Irama Detak Jantung

Jantung (dalam bahasa Latin yaitu *cor*) adalah rongga organ berotot yang memiliki tugas untuk memompa darah melalui pembuluh darah oleh kontraksi berirama yang terjadi secara berulang. Jantung terletak di dalam rongga dada agak sebelah kiri, di antara paru-paru kiri dan paru-paru kanan. Istilah yang terkait dengan jantung, dari kata Yunani *cardia*. Jantung adalah salah satu organ tubuh manusia yang berperan dalam sistem peredaran darah (Dudung, 2014). Jantung akan menyediakan oksigen darah yang cukup pada tubuh dan dialirkan ke seluruh tubuh, serta membersihkan tubuh dari hasil metabolisme (karbondioksida). Sehingga untuk melaksanakan fungsi tersebut jantung akan mengumpulkan darah yang kekurangan oksigen dan kemudian akan memompanya ke paru-paru, dengan cara darah pada jantung akan mengambil oksigen dan kemudian akan membuang karbondioksida (Tulus W, 2015).

Dibawah ini merupakan gambar dan bagian-bagian dari jantung yang ditunjukkan pada Gambar 2.3.



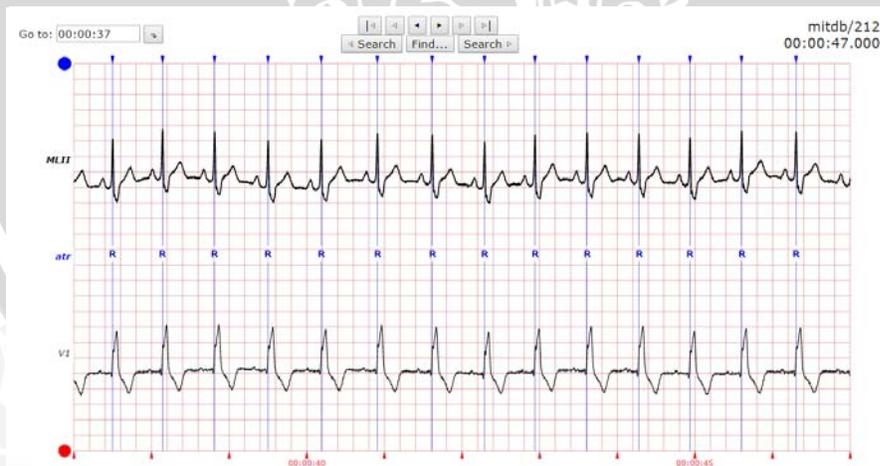
Gambar 2.3 Bagian Jantung.
Sumber : Wayan Tulus (2015).

Jantung umumnya dianggap memiliki sisi kanan dan kiri. Jantung terdiri dari empat ruang, dua di sebelah kanan dan dua di sebelah kiri. Ruang-ruang yang dikenal sebagai atrium dan ventrikel. Setiap sisi jantung terdiri dari satu atrium dan satu ventrikel. Atrium adalah ruang menerima jantung yang menerima darah mengalir kembali ke jantung. Ventrikel adalah bilik jantung yang memompa darah keluar dari jantung. Katup jantung terletak di dalam bilik jantung dan sangat penting dalam aliran darah yang tepat menuju jantung (*USC Cardiothoracic Surgery*, 2016).

2.3 Normal

Jantung yang normal akan memberikan gambaran rekaman dengan pola tertentu. Pola rekaman yang tidak normal memberi petunjuk adanya kelainan jantung (JJ, 2013). Denyut jantung yang normal merupakan tanda kesehatan pada tubuh yang baik secara keseluruhan.

Detak jantung yang normal akan terasa lebih cepat berdetak atau lebih lambat tergantung dari aktifitas yang sedang dikerjakan. Misalnya saat beraktifitas berat maka jantung akan berdetak dengan cepat dan sebaliknya saat beraktifitas ringan akan berdetak dengan lambat. Rata-rata detak jantung selama istirahat dapat dibagi menjadi empat kelompok usia, yaitu bayi berusia di bawah 1 tahun, anak berusia 1-10 tahun, anak remaja berusia 11-17 tahun, dan usia dewasa di atas 17 tahun. Untuk usia <1 tahun, maka detak jantung normal sekitar 100-160 per menit, usia 1-10 tahun, detak jantung normal sekitar 70-120 per menit, usia 11-17 tahun, detak jantung normal sekitar 60-100 per menit, dan usia >17 tahun, maka detak jantung normal sekitar 60-100 per menit (PT. Deherba Indonesia, 2015).



Gambar 2.4 Hasil Pemeriksaan EKG Kondisi Normal.

Sumber : MIT-BIH Lightwave (2010).

Pada Gambar 2.4 di atas merupakan hasil pemeriksaan EKG untuk salah satu kondisi ritme normal sinus. Data-data tersebut yaitu merupakan perempuan dengan umur 32 tahun. Data tersebut di lihat dari dua sisi pemeriksaan yaitu *MLII* dan *V1* (MIT-BIH Records 212, 2010).

2.4 Aritmia

Aritmia Jantung (juga disebut sebagai Disritmia) adalah kondisi jantung dengan ritme atau detak yang tidak normal yang bukan disebabkan dari bagian tubuh yang sama, melainkan dari bagian lain dari tubuh kita (dalam bentuk *sinus node*). Istilah Jantung Berdebar (*palpitations*) mengacu pada sebuah sensasi yang dirasakan oleh seseorang jika yang bersangkutan terkena gejala aritmia jantung (*cardiac arrhythmia*) (The Harley Street Clinic, 2012). Berdasarkan namanya, aritmia (a yaitu tidak, ritmia yaitu ritmik atau irama). Sehingga aritmia (gangguan irama jantung) adalah kondisi di mana irama jantung tidak normal atau bisa dikatakan kelainan denyut jantung. Pada kasus ini penderita bisa memiliki irama jantung yang terlalu cepat atau lambat yang tidak sesuai dengan aktifitas yang sedang dikerjakan (Sindo, 2010).

Bagi kebanyakan orang, denyut jantung yang normal berkisar antara 50 sampai 100 denyut per menitnya. Aritmia jantung (*cardiac arrhythmia*) dapat terjadi pada seseorang dengan denyut jantung yang normal, atau terlalu pelan (dibawah 50 detak per menit) bahkan terlalu cepat (diatas 100 detak per menit). Selain itu dalam merasakan gejala jantung berdebar, seseorang dengan kondisi aritmia juga mungkin mengalami gejala lainnya seperti rasa pening, rasa lelah, sesak nafas serta hilangnya kesadaran secara tiba-tiba. Jenis-jenis pemeriksaan yang bisa mengenali aritmia adalah pemeriksaan elektrokardiogram, pengawasan holter 24 jam, perekaman kondisi, tes latihan *treadmill*, dan pemeriksaan elektrofisiologi (The Harley Street Clinic, 2012). Jenis-jenis aritmia bisa di lihat pada penjelasan di bawah ini.

2.4.1 Atrial Fibrillation



Gambar 2.5 Hasil Pemeriksaan EKG Kondisi Atrial fibrillation.

Sumber : MIT-BIH Lightwave (2010).

Fibrilasi bilik jantung atrium (*Atrial fibrillation (AF)*) merupakan suatu kondisi ritme jantung tidak normal yang paling umum dan menyerang jutaan manusia di seluruh dunia. Kondisinya meliputi aktifitas yang sangat cepat dan tidak beraturan dalam atria dan menimbulkan gejala-gejala seperti jantung

berdebar, sakit kepala, kehilangan kesadaran, sesak nafas dan rasa letih. Fibrilasi bilik jantung atrium (AF) juga merupakan penyebab umum dari stroke yang rata-rata menyerang 5% dengan fibrilasi bilik jantung atrium (AF) setiap tahunnya. Walaupun hal ini mungkin saja menyerang orang muda, penyakit ini menjadi semakin umum dengan bertambahnya umur dan biasanya menyerang pasien yang lebih tua diatas umur 65 tahun / lebih (The Harley Street Clinic, 2012).

Pada Gambar 2.5 di atas merupakan hasil pemeriksaan EKG untuk salah satu kondisi ritme *Atrial fibrillation*. Data-data tersebut yaitu merupakan perempuan dengan umur 80 tahun. Data tersebut di lihat dari dua sisi pemeriksaan yaitu *MLII* dan *V1* (MIT-BIH Records 228, 2010).

2.4.2 PVC Bigeminy

PVC Bigeminy adalah terjadinya *PVC* pada setiap detaknya. *PVC* (*Premature Ventricular Contractions*) atau ventrikel kontraksi prematur adalah denyut jantung yang terjadi sebelum waktunya yang berasal dari ventrikel jantung. Pada *ventricular bigeminy* keadaan *PVC* lebih sering dialami dan dinyatakan sedikit berbeda dari jenis *PVC* lainnya. Sehingga jika perubahan terbaru dalam irama detak jantung terjadi, maka harus dilakukan pemeriksaan jantung dan fungsi hati serta tes darah (Dr. Tryzelaar, 2016).



Gambar 2.6 Hasil Pemeriksaan EKG Kondisi *PVC Bigeminy*.

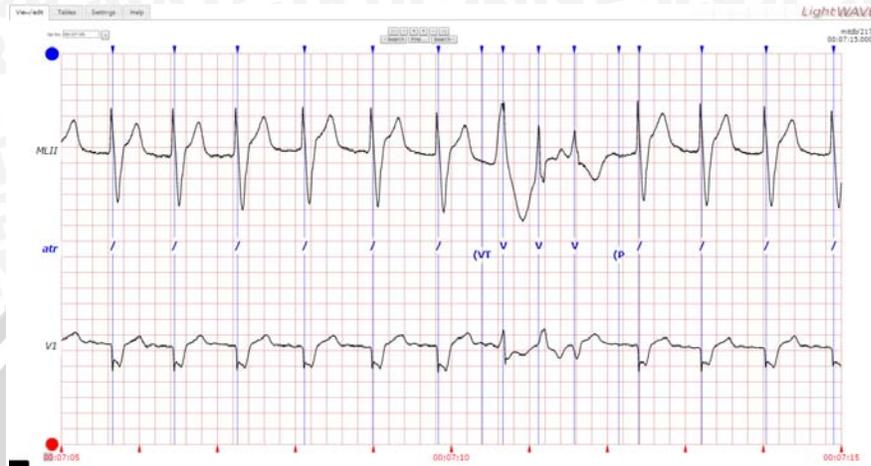
Sumber : MIT-BIH Lightwave (2010).

Pada Gambar 2.6 di atas merupakan hasil pemeriksaan EKG untuk salah satu kondisi ritme *PVC Bigeminy*. Data-data tersebut yaitu merupakan perempuan dengan umur 24 tahun. Data tersebut di lihat dari dua sisi pemeriksaan yaitu *MLII* dan *V1* (MIT-BIH Records 106, 2010).

2.4.3 Ventricular Tachycardia

Takikardia adalah denyut jantung yang terjadi lebih cepat daripada denyut jantung normal. Jantung orang dewasa yang sehat biasanya berdetak selama 60 sampai 100 kali per menit ketika sedang beristirahat. Denyut jantung dikendalikan oleh sinyal-sinyal listrik yang dikirim ke seluruh jaringan jantung. Takikardia disebabkan oleh suatu kelainan di dalam jantung sehingga

menghasilkan sinyal listrik yang cepat. Dalam beberapa kasus, takikardia tidak menimbulkan komplikasi. Namun, takikardia yang parah dapat mengganggu fungsi normal jantung, meningkatkan resiko stroke, atau menyebabkan serangan jantung mendadak atau kematian (DetikHealth.com, 2015).



Gambar 2.7 Hasil Pemeriksaan EKG Kondisi *Ventricular Tachycardia*.

Sumber : MIT-BIH Lightwave (2010).

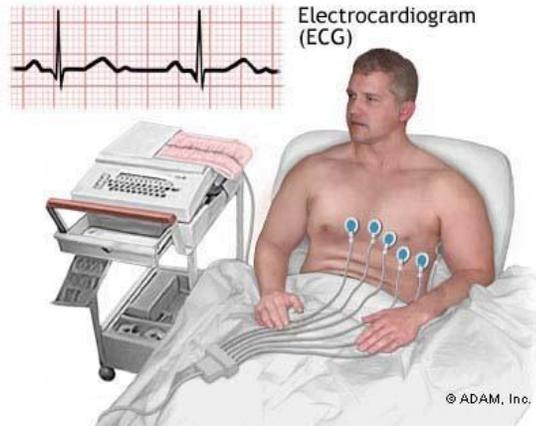
Pada Gambar 2.7 di atas merupakan hasil pemeriksaan EKG untuk salah satu kondisi ritme *Ventricular Tachycardia*. Data-data tersebut yaitu merupakan laki-laki dengan umur 65 tahun. Data tersebut di lihat dari dua sisi pemeriksaan yaitu *MLII* dan *V1* (MIT-BIH Records 217, 2010).

2.5 Pemeriksaan EKG (Elektrokardiografi)

Pemeriksaan EKG (elektrokardiografi) adalah pemeriksaan penunjang jantung yang tertua, sejak permulaan abad ke-20. Walaupun pemeriksaan ini merupakan pemeriksaan yang sudah lama ada, tetapi hingga sekarang pemeriksaan EKG masih merupakan pemeriksaan yang penting, dan tak tergantikan dengan pemeriksaan-pemeriksaan lain yang lebih baru (JJ, 2013). Rekaman listrik jantung yang dihasilkan dari EKG dapat memberi petunjuk adanya beberapa kelainan jantung seperti Gangguan irama jantung, penyakit jantung koroner, serangan jantung, penebalan otot jantung dan pembesaran rongga jantung (JJ, 2013). Pengertian Elektrokardiografi sesuai namanya 'grafi' merupakan pemeriksaan atau pencatatan, elektro merupakan aktivitas listrik, dan kardio merupakan jantung sehingga jika digabungkan menjadi pemeriksaan aktivitas listrik jantung atau elektrokardiografi.

Rekaman EKG sangat bermanfaat dalam memberikan informasi tentang berbagai kelainan jantung. Tetapi, tidak semua kondisi jantung bisa dinilai dengan EKG. EKG tidak dapat menilai kemampuan kontraksi atau pompa jantung, EKG juga tidak dapat menentukan ada tidaknya kebocoran katup atau sekat jantung, EKG juga tidak dapat menentukan ada tidaknya penyempitan katup jantung. Apalagi menentukan berat ringannya kebocoran atau penyempitan katup jantung (JJ, 2013). Pada Gambar 2.8 di bawah ini merupakan gambar pada

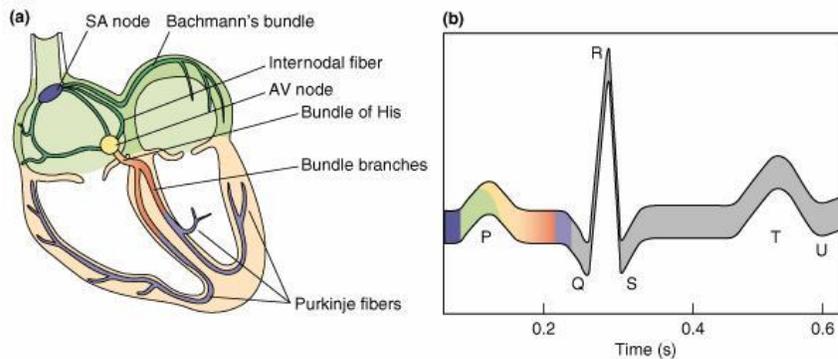
pemeriksaan EKG dimana pasien berbaring nyaman mungkin kemudian dipasang 10 elektroda, 6 buah di dada, 2 buah di lengan, dan 2 buah di tungkai.



Gambar 2.8 Alat Pemeriksaan EKG.

Sumber : JJ (2013).

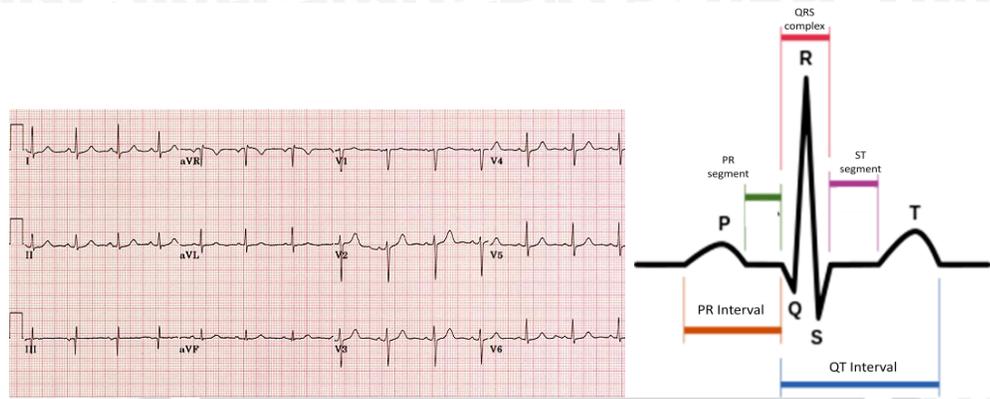
Aktivitas jantung yang dimulai dari keadaan istirahat, kemudian kontraksi atrium, disusul kontraksi ventrikel, dan kembali ke keadaan istirahat merupakan suatu siklus yang terjadi secara berulang terus menerus sepanjang hidup. Aktivitas kelistrikan yang mengatur siklus kerja jantung ini dapat direkam dengan menggunakan alat yang disebut elektrokardiograf, dan hasil rekamannya disebut elektrokardiogram yang disingkat EKG atau *ECG* (Ponco Siwindarto, 2012). Di bawah ini merupakan hasil rekaman EKG dalam satu siklus jantung.



Gambar 2.9 Rekaman EKG dalam Siklus Jantung.

Sumber : Ponco Siwindarto (2012).

Dalam hasil rekaman EKG, satu siklus jantung terdiri atas beberapa gelombang, yaitu gelombang P, Q, R, S, T, dan U. Gelombang-gelombang tersebut berhubungan dengan aktivitas listrik yang terjadi di dalam jantung, dimana pada Gambar 2.9 ditunjukkan oleh warna yang sama antara gambar bagian (a) dan gambar bagian (b). Gelombang P ditimbulkan oleh depolarisasi atrium; gelombang Q, R, dan S yang bersama-sama membentuk kompleks QRS ditimbulkan oleh depolarisasi ventrikel; dan gelombang T ditimbulkan oleh repolarisasi ventrikel. Gelombang U kemungkinan ditimbulkan oleh repolarisasi serabut Purkinje (Ponco Siwindarto, 2012).

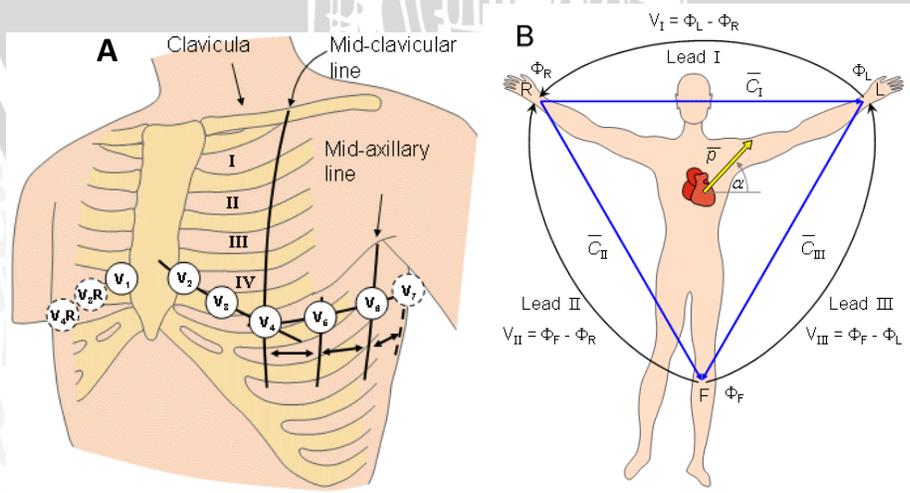


Gambar 2.10 EKG Normal.
 Sumber : JJ (2013).

Pada Gambar 2.10 di atas merupakan gambar hasil dari EKG (Elektrokardiaografi) dalam keadaan yang normal. Tampak dari Gambar 2.10 tersebut semua gelombang yaitu P, Q, R, S, dan T menunjukkan hasil yang normal. Dimana gelombang P terlihat ada, kemudian terdapat gelombang Q, R, S, dan T. Pada gambar di atas pun terlihat contoh dari kertas EKG yang biasanya digunakan dalam hasil pemeriksaan EKG.

2.6 Sadapan EKG

Dalam pemeriksaan elektrokardiografi (EKG), untuk mendapatkan hasil pemeriksaan yang lebih akurat dibutuhkan 12 sadapan hasil rekam EKG. 12 sadapan itu antara lain I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, V6. Alasan utama memakai 12 sadapan untuk pemeriksaan EKG adalah untuk meningkatkan pengenalan pola sehingga dapat memberikan hasil yang lebih akurat dengan membandingkan hasil vector dari sudut yang berbeda (Malmivuo, J., Plonsey, R., 1995). Pada Gambar 2.11 yang menunjukkan hasil rekam EKG normal dengan 12 sadapan.



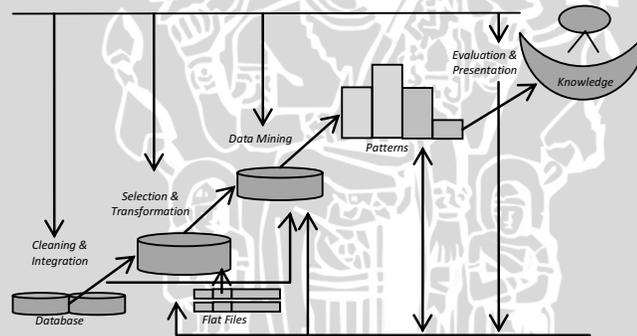
Gambar 2.11 Sadapan EKG.
 Sumber : Malmivuo, J., Plonsey, R., 1995.

Pada penelitian ini digunakan sadapan dari kanal *MLII* dan *V1*. *MLII* atau *modified lead II* merupakan penempatan elektroda di lengan kanan dan tungkai kiri, dengan tungkai kiri positif. *V1* digunakan untuk memeriksa sinyal rendah, dan untuk sinyal atas elektroda juga ditempatkan di dada. *V1* merupakan sadapan pada sela iga keempat sebelah kanan dari *sternum* seperti yang terlihat pada Gambar 2.11 di atas (Moody, G.B., 2010).

2.7 Data Mining

Data mining merupakan sekumpulan proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan. Kata *mining* sendiri berarti usaha untuk mendapatkan pengetahuan dari suatu data. Oleh karena itu *data mining* sebenarnya memiliki akar yang panjang dari bidang ilmu seperti kecerdasan buatan (*artificial intelligent*), *machine learning*, statistik dan basis data. Adapun beberapa teknik yang ada dalam literatur *data mining* antara lain yaitu *association rule mining*, *clustering*, klasifikasi, *neural network*, *genetic algorithm* dan lain-lain (Kusnawi, 2007).

Data Mining didefinisikan sebagai penggalian informasi dari himpunan data yang sangat besar dan dapat dikatakan merupakan sebuah prosedur pengetahuan *mining* dari data (Tutorials Point (I) Pvt.ltd, 2014). Dibawah ini merupakan tahapan dalam *data mining*.



Gambar 2.12 Tahapan Data Mining.

Sumber : Tutorials Point (I) Pvt.LTD (2014).

Tahapan dalam proses *data mining* seperti Gambar 2.12 di atas yaitu melakukan proses pembersihan data dari *database* dengan membuang data yang tidak konsisten dan terdapat *noise*. Selain itu juga melakukan integrasi data karena data didapat tidak hanya berasal dari satu *database* saja tetapi berasal dari beberapa *database* atau *file* teks. Sehingga pada langkah ini menggabungkan beberapa data menjadi satu *database*. Transformasi dan pemilihan data ini untuk menentukan kualitas dari hasil *data mining*, sehingga data diubah menjadi bentuk yang sesuai untuk di-Mining. Kemudian menuju ke *data mining* dimana pada langkah ini yaitu metode cerdas yang diterapkan dalam rangka untuk mengekstrak pola-pola data. Kemudian pola data yang ada dievaluasi untuk menilai apakah hipotesa yang ada memang tercapai. Dari hasil itulah pola direpresentasikan menjadi pengetahuan (Tutorials Point (I) Pvt.ltd, 2014).

2.7.1 Normalisasi

Pada tahapan *data mining* seperti pada Gambar 2.12 di atas terdapat proses transformasi data. Metode yang bisa digunakan dalam proses transformasi salah satunya adalah proses normalisasi data. Normalisasi melibatkan skala semua nilai untuk atribut yang diberikan agar nilai antar fitur memiliki selisih yang tidak jauh atau kisaran nilai yang kecil. Normalisasi digunakan ketika dalam tahap belajar, jaringan saraf atau metode yang melibatkan pengukuran digunakan (Tutorials Point (I) Pvt.ltd, 2014).

Metode normalisasi yang digunakan adalah *Min-Max Normalization*. *Min-max normalization* bekerja dengan melihat seberapa banyak nilai dari nilai minimum $\min(X)$ dan skala perbedaan ini dengan kisaran. Adapun rumus untuk metode ini ditunjukkan pada Persamaan 2.1 (Daniel T. Larose, 2005).

$$X^* = \left(\frac{X - \min(X)}{\text{range}(X)} \right) = \left(\frac{X - \min(X)}{\max(X) - \min(X)} \right) \quad (2.1)$$

Keterangan :

X^* = Nilai yang telah dinormalisasi.

X = Nilai atau data yang ingin dinormalisasi.

$\min(X)$ = Mencari nilai minimal atau nilai terkecil dari data X .

$\max(X)$ = Mencari nilai maksimal atau nilai terbesar dari data X .

Langkah-langkah yang dilakukan dalam proses normalisasi data latih dan data uji yaitu sebagai berikut.

1. Masukan nilai data latih dan data uji.
2. Mencari nilai minimum dan maksimum dari tiap fitur atau parameter yang terdapat pada data latih dan data uji.
3. Melakukan perhitungan dengan menggunakan *Min-Max Normalization*.

Output dari hasil perhitungan merupakan data data latih dan data uji yang telah ternormalisasi.

2.7.2 Machine Learning

Machine Learning adalah ilmu dalam bidang komputer untuk bertindak tanpa eksplisit diprogram. Dalam dekade terakhir, pembelajaran mesin telah memberi kita keuntungan bahwa kita bisa membuat sendiri, pengenalan suara praktis, pencarian web yang efektif, dan pemahaman yang jauh lebih baik dari genom manusia. Banyak peneliti juga berpikir itu adalah cara terbaik untuk membuat kemajuan menuju tingkat AI manusia (Stanford University, 2015).

Dua metode pembelajaran mesin yang paling banyak diadopsi adalah *supervised learning* (Data memiliki label atau kelas) dan *unsupervised learning* (Data tidak memiliki label atau kelas). Perbedaan *machine learning* dengan *data mining* adalah *data mining* menemukan pola yang sebelumnya tidak diketahui

sedangkan pembelajaran mesin digunakan untuk mereproduksi pola dan pengetahuan yang telah dikenal dan kemudian secara otomatis menerapkan hasil tersebut untuk pengambilan keputusan dan tindakan (SAS Institute Inc, 2014).

2.7.3 Klasifikasi

Klasifikasi adalah salah satu aspek dalam *data mining*. Teknik klasifikasi juga telah banyak digunakan pada berbagai permasalahan dalam suatu penelitian. Klasifikasi merupakan proses untuk menemukan model yang menggambarkan kelas data atau konsep dengan tujuan agar dapat memprediksi kelas dari objek yang label kelas tidak diketahui. Dimana dapat menemukan model yang berasal yang menggambarkan dan membedakan kelas data atau konsep. Model yang Berasal didasarkan pada analisis himpunan data pelatihan yaitu objek data yang label kelasnya diketahui (Tutorials Point (I) Pvt.ltd, 2014).

Klasifikasi adalah proses untuk menentukan suatu kelompok pada setiap keadaan. Setiap keadaan berisi sekelompok atribut, salah satunya adalah *class attribute*. Sedangkan klustering digunakan untuk mengidentifikasi kelompok alami dari sebuah kasus yang di dasarkan pada sebuah kelompok atribut, mengelompokkan data yang memiliki kemiripan atribut, klustering tidak memiliki kelas awal sehingga tugas kita untuk mencari kemiripan data (Dudung, 2015).

Adapun beberapa algoritma klasifikasi, antara lain *Naive Bayes (NB)*, *Support Vector Machine (SVM)*, *Artificial Neural Networks (ANN)*, *Decision Tree Induction*, *Genetic Algorithms* (Tutorials Point (I) Pvt.ltd, 2014 dan Chandani V, dkk, 2015).

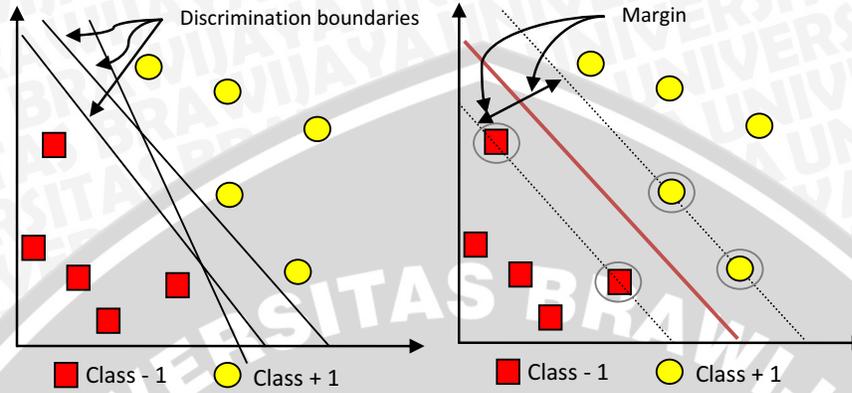
2.8 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah sebuah metode klasifikasi biner yang dikembangkan oleh Vapnik dan rekan-rekannya di Laboratorium Bell, yang merupakan algoritma lanjutan pada perbaikan yang dilakukan pihak lain (Gjorgji Madzarov, Dejan Gjorgjeviki, dkk, 2008). *SVM* ini dibatasi untuk masalah biner, dan terdapat titik data *training* seperti pada Persamaan 2.2 di bawah ini.

$$\{X_i, y_i\}, i = 1, \dots, l, y_i \in \{-1, 1\}, x_i \in R^d \quad (2.2)$$

Data yang tersedia dinotasikan dengan $x_i \in R^d$ sedangkan label atau kelas masing-masing dinotasikan dengan $y_i \in \{-1, 1\}$ untuk $i=1, 2, \dots, l$. dimana l adalah banyaknya data. Dalam metode ini, kita berusaha untuk menentukan fungsi *hyperplane* atau klasifier terbaik antara fungsi yang tidak terbatas jumlahnya untuk memisahkan dua macam objek. *Hyperplane* terbaik adalah *hyperplane* yang terletak di tengah-tengah antara himpunan objek dari dua kelas (kelas positif dan kelas negatif), mencari *hyperplane* terbaik ekuivalen dengan mencari *margin* yang maksimal atau jarak antara dua objek dari kelas yang berbeda (Zaki Imaduddin & Hilmy Abidzar T., 2015). Gambar 2.13 merupakan pemisahan *hyperplane* yang paling optimal atau terbaik. Dimana kotak merah menggambarkan kelas negatif (-1) dan bulat kuning menggambarkan kelas positif

(1). Kemudian mendapatkan *hyperplane* pemisah yang ditunjukkan dengan garis miring yang memisahkan dua kelas tersebut. Untuk Gambar 2.13 merupakan SVM dengan sebaran data secara linear.



Gambar 2.13 Pemisahan *Hyperplane* Terbaik.

Sumber : Anto Satrio N., dkk (2003).

2.8.1 SVM Linear

Fungsi klasifikasi *Support Vector Machine* linear digunakan untuk memisahkan data pelatihan menjadi dua kelas dengan memisahkan *hyperplane* yang digambarkan dengan sebaran data secara linear (Zuyu Yin & Jian Hou, 2016). Konsep *Support Vector Machine* bermula dari masalah klasifikasi dua kelas seperti *training set* positif dan negatif. Dalam persamaan di bawah biasanya terdapat dua kelas sebagai nilai *output* dari klasifikasi ini yaitu nilai -1 dan 1. Untuk kasus terpisah secara linear, algoritma SVM hanya terlihat untuk *hyperplane* yang memisahkan dengan margin terbesar. Ini dapat dirumuskan pada Persamaan 2.3 dan 2.4 sebagai berikut (Gjorgji Madzarov, Dejan Gjorgjevikj, dkk, 2008).

$$x_i \cdot w + b \geq 1, \text{ untuk } y_i = 1 \tag{2.3}$$

$$x_i \cdot w + b \leq -1, \text{ untuk } y_i = -1 \tag{2.4}$$

Keterangan :

x_i = Data ke-i.

w = bobot *super vector* atau *vector* yang tegak lurus terhadap *hyperplane*.

b = nilai bias.

y_i = kelas data ke-i.

Dari dua persamaan di atas maka dapat dikombinasikan sehingga mendapatkan *hyperplane* yaitu yang terdapat pada Persamaan 2.5 sebagai berikut (Gjorgji Madzarov, Dejan Gjorgjevikj, dkk, 2008).

$$y_i(x_i \cdot w + b) - 1 \geq 0, \text{ untuk } i = 1, \dots, n. \tag{2.5}$$

Keterangan :

n = jumlah data.

Hyperplane terbaik merupakan *hyperplane* yang terletak pada tengah-tengah antara dua *set* objek dari dua kelas. *Hyperplane* terbaik dapat ditemukan dengan mengukur margin terbesar dari beberapa alternatif *hyperplane* yang ada. *Margin* adalah jarak terdekat antara dua *set* objek dari dua kelas. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Garis solid pada Gambar 2.13 pada gambar ke 2 menunjukkan *hyperplane* yang terbaik, yaitu yang terletak tepat pada tengah-tengah kedua kelas, sedangkan titik merah dan kuning yang berada dalam lingkaran hitam adalah *support vector*. Proses untuk mendapatkan lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM (Anto Satrio N., dkk, 2003).



Gambar 2.14 Linear memisahkan *hyperplanes* untuk kasus terpisah.

Sumber : Gjorgji Madzarov, Dejan Gjorgjevikj, dkk (2008).

Pada Gambar 2.14 di atas menjelaskan pemisahan *hyperplane* secara linear untuk kasus yang terpisah. Perhatikan bahwa H_1 dan H_2 sejajar (mereka memiliki nilai yang sama normal) dan tidak ada poin *training* jatuh di antara mereka. Dengan demikian kita dapat menemukan sepasang *hyperplanes* yang memberikan margin maksimal dengan memaksimalkan fungsi $\frac{1}{\|w\|}$ atau dengan meminimalkan $\|w\|^2$.

Permasalahan SVM linear dapat diselesaikan dengan mendapatkan *hyperplane* dengan menggunakan fungsi pada Persamaan 2.6 (Cholissodin I, dkk, 2014).

$$f(x) = (w \cdot x) + b \tag{2.6}$$

Kita definisikan dengan Persamaan 2.7.

$$w = \sum_{i=1}^n \alpha_i y_i x_i \tag{2.7}$$

dan Persamaan 2.8.

$$b = -\frac{1}{2}(w \cdot x^+ + w \cdot x^-) \tag{2.8}$$

x^+ dan x^- adalah data yang merupakan *support vector* dari kelas positif dan negatif yang memiliki nilai bobot (α) tertinggi di setiap kelas dan w adalah bobot *support vector*. Fungsi keputusan adalah $sign(f(x))$, pada Persamaan 2.9.

$$f(x) = \sum_{i=1}^m \alpha_i y_i (x \cdot x_i) + b \tag{2.9}$$

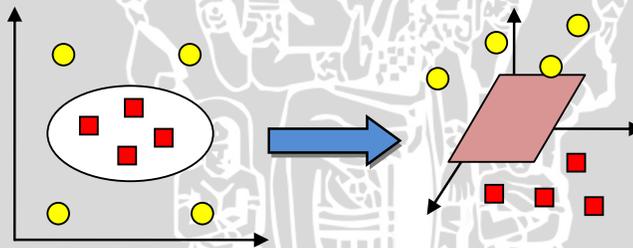
Parameter n dinotasikan sebagai banyaknya data dan m adalah jumlah bilangan dari *support vector*. Fungsi tersebut dapat menentukan yang mana data tersebut yang menempati kelas positif maupun kelas negatif pada Persamaan 2.10 dan 2.11.

$$sign(f(x)) = 1 \text{ untuk kelas positif} \tag{2.10}$$

$$sign(f(x)) = -1 \text{ untuk kelas negatif} \tag{2.11}$$

2.8.2 SVM Non-Linear

Pada umumnya permasalahan dalam ruang lingkup dunia nyata (*real world problem*) jarang terdapat permasalahan yang bersifat linear, tetapi kebanyakan bersifat *non-linear*. Untuk menyelesaikan permasalahan *non linear*, maka memodifikasi SVM dengan memasukkan fungsi *kernel* kedalam algoritma SVM (Anto Satrio N., dkk, 2003).



Gambar 2.15 Data *Hyperplane* Non-Linear.

Sumber : Anto Satrio N., dkk (2003).

Non-linear selalu disebabkan oleh *nonlinearities* atau kebisingan, yang merupakan sumber data *non-linear*. Pada Gambar 2.15 di atas merupakan contoh dengan *hyperplane* pada kasus *non-linear*. Klasifikasi *hyperplane* umum adalah nilai minimum berdasarkan Persamaan 2.12 di bawah ini (Zuyu Yin & Jian Hou, 2016).

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l \xi_i \tag{2.12}$$

Keterangan:

w = bobot *support vector*.

ξ_i = variabel *slack* (mengukur *error* dari data).

- C = nilai *complexity* bernilai bilangan positif (batasan *error*).
 l = jumlah data.

Nilai C (*Complexity*) bertujuan untuk meminimalkan *error* dan memperkecil nilai variabel *slack*. Karena keterbatasan *SVM*, maka kemudian dikembangkan fungsi kernel untuk mengklasifikasikan data dengan bentuk *non-linear*. Tabel 2.2 di bawah ini merupakan beberapa macam fungsi kernel *SVM*.

Tabel 2.2 Fungsi Kernel SVM.

No.	Fungsi Kernel	Formula
1	<i>Linear</i>	$k(x_i, y_i) = x_i \cdot x_j$
2	<i>Invers Multiquadric</i>	$k(x, y) = \frac{1}{\sqrt{\ x - y\ ^2 + (c)^2}}$
3	<i>Laplacian Kernel</i>	$k(x, y) = \exp\left(-\frac{\ x - y\ }{\sigma}\right)$
4	<i>Cauchy Kernel</i>	$k(x, y) = \frac{1}{1 + \frac{\ x - y\ ^2}{\sigma^2}}$
5	<i>Polynomial Degree Kernel</i>	$K(X_i, X_j) = (X_i \bullet X_j)^d$
6	<i>Gaussian Radial Basis Function</i>	$k(x_i, x_j) = \exp\left(-\frac{\ x_i - x_j\ ^2}{2\sigma^2}\right)$

Sumber : Zuyu Yin & Jian Hou (2016), Souza, Cesar (2010), Subhransu Maji, dkk (2012) dan Dr. Saed Sayad (2010) .

Setelah menyelesaikan perhitungan dengan salah satu kernel di atas. Hasil dari perhitungan kernel kemudian akan digunakan dalam *sequential training SVM*. Untuk mendapatkan nilai $f(x)$ yang digunakan pada data uji dengan Persamaan 2.13 di bawah ini (Cholissodin I, dkk, 2014).

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x, x_i) + b \tag{2.13}$$

2.8.3 Sequential Training SVM

Metode *Quadric Programming (QP)* merupakan salah satu cara yang biasa digunakan untuk mencari *hyperplane* terbaik atau yang paling optimal pada permasalahan klasifikasi menggunakan metode *SVM*. Tetapi penyelesaian *QP* biasanya cukup kompleks, membuang-buang waktu, dan rentan terhadap ketidakstabilan numerik. Kemudian terdapat metode *sequential training SVM* yang digunakan sebagai alternatif sederhana untuk menemukan *hyperplane* optimal. Algoritmanya yaitu sebagai berikut (Cholissodin I, dkk, 2014).

1. Inisialisasi $\alpha_i = 0$ dan parameter lain, misalnya $\lambda = 2$, konstanta $\gamma = 2$, $C=1$, Iterasi Maksimum = 10, dan $\epsilon = 0.0001$. Kemudian menghitung matriks *Hessian* menggunakan Persamaan 2.14.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \text{ untuk } i, j = 1, \dots, l \quad (2.14)$$

Keterangan :

x_i = data ke- i

x_j = data ke- j

y_i = kelas data ke- i

y_j = kelas data ke- j

l = jumlah data

$K(x_i, x_j)$ = fungsi kernel yang digunakan.

2. Kemudian memulai proses iterasi mulai dari data ke- i sampai ke- j , untuk setiap pola $i=1$ sampai ke- l , dengan menggunakan Persamaan 2.15, 2.16, dan 2.17 di bawah ini.

$$E_i = \sum_{j=1}^l \alpha_j D_{ij} \quad (2.15)$$

$$\delta \alpha_i = \min \{ \max [\gamma (1 - E_i), -\alpha_i], C - \alpha_i \} \quad (2.16)$$

$$\alpha_i = \alpha_i + \delta \alpha_i \quad (2.17)$$

Keterangan :

γ = learning rate = konstanta $\gamma / \max_{ij} D_{ij}$.

$\max_{ij} D_{ij}$ = Nilai maksimum dari diagonal matriks *hessian*.

3. Ulangi langkah ke-2 hingga kondisi iterasi maksimum telah tercapai atau $\max (|\delta \alpha_i|) < \epsilon$.
4. Kemudian didapatkan nilai *support vector (sv)*, $SV = (\alpha_i > \text{threshold})$. Nilai *threshold sv* didapatkan dari beberapa percobaan, biasanya digunakan $\text{threshold} \geq 0$.

2.8.4 Karakteristik SVM

Karakteristik atau ciri dari metode *Support Vector Machine (SVM)* yaitu sebagai berikut (Anto Satrio N., dkk, 2003).

1. Secara prinsip SVM adalah linear *classifier*, tetapi kita bisa menambahkan fungsi kernel jika permasalahan termasuk *non-linear*.
2. Pada metode SVM, salah satu keunggulannya merupakan suatu metode dimana baik dilakukan jika data memiliki parameter atau fitur atau input *space* ke ruang yang berdimensi tinggi, dan optimasi dilakukan pada ruang vektor yang baru. Hal ini membedakan SVM dari solusi pengenalan pola pada

umumnya, yang melakukan optimasi parameter pada ruang hasil transformasi yang berdimensi lebih rendah daripada dimensi *input space*.

3. Menerapkan strategi *Structural Risk Minimization (SRM)* dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah kelas pada *input space*.
4. Prinsip kerja pada dasarnya hanya mampu menangani klasifikasi dua kelas.

2.8.5 Kelebihan dan Kekurangan SVM

Metode SVM juga memiliki kelebihan dan kelemahan dibandingkan metode konvensional lain. Kelebihan SVM antara lain sebagai berikut (Anto Satrio N., dkk, 2003).

1. Generalisasi

Generalisasi didefinisikan sebagai kemampuan suatu metode (SVM, *neural network*, dsb.) untuk mengklasifikasikan suatu pola, yang tidak termasuk data yang dipakai dalam fase pembelajaran metode itu. Generalisasi dipengaruhi oleh dua faktor yaitu *error* terhadap *training set* dan dipengaruhi oleh dimensi VC (Vapnik-Chervokinensis). SVM meminimalkan kedua *error* tersebut. Strategi ini disebut *SRM*.

2. *Curse of dimensionality*

Metode SVM merupakan salah satu metode yang tepat dipakai untuk memecahkan masalah berdimensi tinggi dalam keterbatasan sampel data yang ada. Karena untuk mendapatkan data yang banyak dan bervariasi membutuhkan biaya dan waktu yang besar pula.

3. Feasibility

Metode SVM dapat diimplementasikan relatif mudah, karena proses penentuan *support vector* dapat dirumuskan dalam *QP problem*. Oleh karena itu jika kita memiliki *library* untuk menyelesaikan *QP problem*, maka SVM akan mudah diimplementasikan.

Disamping kelebihannya, SVM juga memiliki kelemahan atau keterbatasan antara lain sebagai berikut (Anto Satrio N., dkk, 2003).

1. Sulit dipakai dalam masalah berskala besar. Skala besar dalam hal ini dimaksudkan dengan jumlah sampel yang diolah.
2. Secara teoritis dikembangkan untuk masalah klasifikasi dengan dua kelas. Tetapi SVM telah dimodifikasi sehingga bisa menyelesaikan masalah multi class dengan bantuan salah satu metodenya adalah *Binary Decision Tree*.

2.9 *Binary Tree* dari SVM

Metode ini menggunakan SVM yang dibuat kedalam struktur pohon biner. Sebuah SVM di setiap node dari pohon ini dilatih menggunakan dua kelas. Algoritma kemudian memproses kemungkinan *output* untuk mengukur kesamaan antara sampel yang tersisa dan dua kelas yang digunakan untuk pelatihan. Semua sampel di *node* ditugaskan untuk dua *subnodes* yang berasal

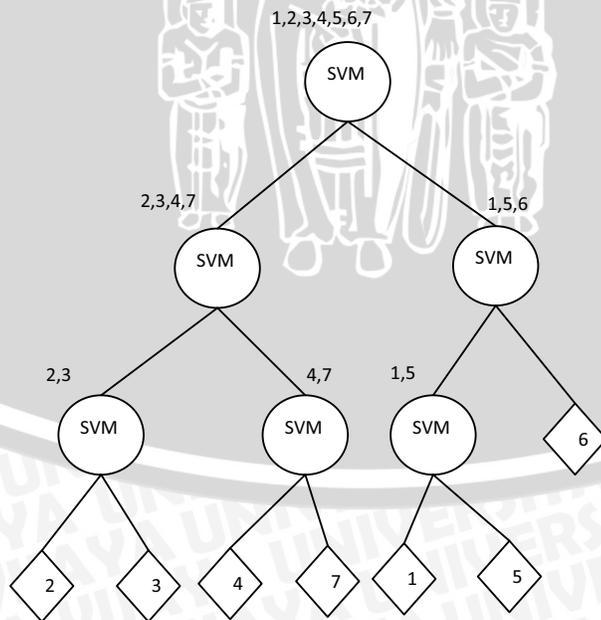
dari kelas yang dipilih sebelumnya. Langkah ini berulang pada setiap simpul sampai setiap node hanya sampel dari satu kelas. Masalah utama yang harus dipertimbangkan serius di sini adalah waktu pelatihan, karena selain pelatihan, kita harus menguji semua sampel di setiap simpul untuk mengetahui kelas berada pada *subnode* yang mana sambil membangun pohon. Ini dapat menurunkan kinerja pelatihan untuk dataset pelatihan besar (Gjorgji Madzarov, Dejan Gjorgjevikj, dkk, 2008).

2.10 SVM-BDT

SVM-BDT memecahkan masalah pengenalan pola multi kelas menggunakan pohon biner, di mana setiap *node* membuat keputusan biner menggunakan klasifikasi *SVM* biner (Felci Rajam dan S. Valli, 2011).

Arsitektur *SVM-BDT* (*Support Vector Machines* memanfaatkan *Binary Decision Tree*), mengambil keuntungan dari kedua perhitungan efisien arsitektur pohon dan akurasi klasifikasi *SVM* yang tinggi. Memanfaatkan arsitektur ini, $N-1$ adalah jumlah level atau panjang *tree* yang terbentuk. Dimana N adalah banyaknya kelas (Gjorgji Madzarov, Dejan Gjorgjevikj, dkk, 2008).

Contoh dari *SVM-BDT* yang memecahkan masalah 7 pola kelas dengan memanfaatkan pohon biner, di mana setiap *node* membuat keputusan biner menggunakan *SVM* ditampilkan pada Gambar 2.16. Setiap sampel dimulai pada akar pohon atau disebut sebagai *root*. Pada setiap *node* dari pohon biner saat sedang dibuat berlaku proses masukan ke dalam salah satu dari dua kelompok yang mungkin diwakili oleh mentransfer apakah pola menuju ke kiri atau ke sub-pohon kanan. Masing-masing kelompok dapat berisi beberapa kelas. Ini diulang secara rekursif hingga bawah pohon sampai sampel mencapai simpul daun yang telah mewakili kelas (Gjorgji Madzarov, Dejan Gjorgjevikj, dkk, 2008).



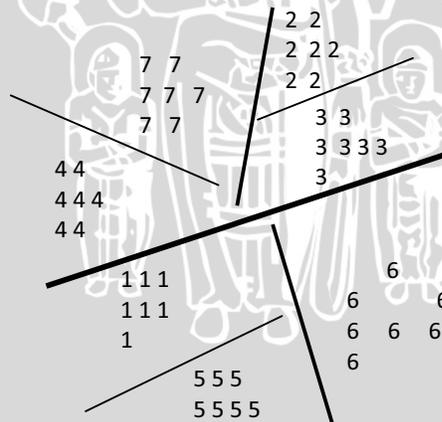
Gambar 2.16 Ilustrasi SVM-BDT.

Sumber : Gjorgji Madzarov, Dejan Gjorgjevikj, dkk (2008).

Untuk konsistensi antara model *clustering* dan cara *SVM* menghitung keputusan *hyperplane*, model pengelompokan menggunakan langkah-langkah jarak pada ruang kernel, bukan di ruang *input*. Karena itu, semua sampel pelatihan dipetakan ke dalam ruang kernel dengan fungsi kernel yang sama yang akan digunakan dalam tahap pelatihan (Gjorgji Madzarov, Dejan Gjorgjevikj, dkk, 2008).

Metode *SVM-BDT* berdasarkan rekursif membagi kelas dalam dua kelompok menguraikan di setiap simpul dari pohon keputusan dan pelatihan *SVM* yang akan memutuskan di mana dari kelompok sampel yang tidak diketahui yang masuk harus ditetapkan (Gjorgji Madzarov, Dejan Gjorgjevikj, dkk, 2008).

Mari kita set sampel x_1, x_2, \dots, x_M masing-masing diberi label $y_i \in \{c_1, c_2, \dots, c_N\}$ di mana N adalah jumlah kelas. Metode *SVM-BDT* dimulai dengan membagi kelas dalam dua kelompok dengan menguraikan g_1 dan g_2 . Hal ini dilakukan dengan menghitung pusat gravitasi (*gravity center*) N untuk N kelas yang berbeda. Kemudian, dua kelas yang memiliki jarak Euclidean terbesar dari satu sama lain ditugaskan untuk masing-masing dua kelompok *clustering*. Setelah ini, kelas dengan jarak Euclidean terkecil dari salah satu kelompok pengelompokan ditemukan dan ditetapkan ke grup yang sesuai. Pusat gravitasi dari kelompok ini dihitung untuk mewakili penambahan sampel dari kelas baru ke grup. Proses berlanjut dengan mencari kelas yang belum ditetapkan berikutnya yang paling dekat dengan salah satu dari kelompok *clustering*, menugaskan ke kelompok yang sesuai dan memperbarui pusat gravitasi kelompok, sampai semua kelas ditugaskan ke salah satu dari dua kelompok mungkin.



Gambar 2.17 Pembagian *SVM-BDT* dalam 7 Kelas.
 Sumber : Gjorgji Madzarov, Dejan Gjorgjevikj, dkk (2008).

Ini mendefinisikan pengelompokan semua kelas dalam dua kelompok menguraikan kelas. Pengelompokan ini kemudian digunakan untuk melatih *classifier SVM* di simpul akar pohon keputusan, menggunakan sampel dari kelompok pertama sebagai contoh positif dan sampel dari kelompok kedua sebagai contoh negatif. Kelas-kelas dari kelompok pengelompokan pertama sedang ditugaskan untuk pertama (kiri) *subtree*, sedangkan kelas kelompok

pengelompokan kedua sedang ditugaskan ke (kanan) *subtree* kedua. Proses ini berlanjut secara rekursif (membagi masing-masing kelompok menjadi dua subkelompok menerapkan prosedur yang dijelaskan di atas), sampai hanya ada satu kelas per kelompok yang mendefinisikan daun di pohon keputusan.

Misalnya, Gambar 2.17 mengilustrasikan pengelompokan 7 kelas, sedangkan Gambar 2.16 menunjukkan pohon keputusan sesuai SVM. Setelah menghitung *gravity center* untuk semua kelas, kelas *c2* dan *c5* ditemukan dengan jarak terjauh dan terpisah satu sama lain, mengingat jarak Euclidean dan ditugaskan untuk *g1* kelompok dan *g2* sesuai. Yang paling dekat dengan *g1* kelompok kelas *c3*, sehingga ditugaskan ke *g1* kelompok, diikuti oleh perhitungan kembali dari pusat gravitasi *g1* ini. Pada langkah berikutnya, kelas *c1* adalah yang paling dekat dengan kelompok *g2*, sehingga ditugaskan untuk kelompok itu dan pusat gravitasi kelompok dihitung ulang. Pada iterasi berikut, kelas *c7* ditugaskan untuk *g1* dan kelas *c6* ditugaskan untuk *g2*, diikuti dengan menghitung ulang pusat gravitasi kelompok. Akhirnya kelas *c4* ditugaskan untuk *g1*. Ini melengkapi putaran pertama pengelompokan yang mendefinisikan kelas yang akan ditransfer ke kiri dan *subtree* kanan dari simpul akar. SVM classifier di root dilatih dengan mempertimbangkan sampel dari kelas {*c2, c3, c4, c7*} contoh positif dan contoh dari kelas {*c1, c5, c6*} contoh negatif.

Langkah-langkah perhitungan algoritma dengan metode BDT-SVM yaitu sebagai berikut (Heena Farooq B. & Mohd Arif W., 2014).

1. Identifikasi label dari data uji dan data latih.
2. Tentukan *center point* atau pusat gravitasi atau *gravity center* dari semua kelas atau label. *Gravity center* adalah titik pusat dari setiap kelas. Untuk mendapatkan titik pusat dari setiap kelas, dapat dihitung dengan mencari rata-rata setiap parameter dari masing-masing setiap kelas. Persamaan untuk menghitung nilai rata-rata bisa dilihat pada Persamaan 2.18 di bawah ini.

$$\overline{GC_{C_i F_j}} = \frac{\sum_{m=1}^n x_m}{n} \tag{2.18}$$

Keterangan:

$\overline{GC_{C_i F_j}}$ = Gravity Center kelas ke-*i* parameter atau fitur ke-*j*, dimana *i*=1,...,jumlah kelas.

x_m = Nilai data ke-*m* pada kelas ke-*i* parameter ke-*j*.

n = jumlah data pada setiap kelas.

3. Hitung jarak dengan menggunakan *Euclidean Distance* diantara dua kelas (*i,j*).
4. Jika jarak euclidean dari pasangan kelas adalah maksimum, maka kedua kelompok kelas didapatkan dengan *i* sebagai kelompok atau kelas pertama (*subtree* kiri) dan *j* sebagai kelompok atau kelas kedua (*subtree* kanan).
5. Menetapkan sisa kelas untuk kelompok terdekat mereka.

6. Mendapatkan *classifier SVM* untuk setiap pasangan kelas.
7. Ulangi prosedur ini sampai simpul daun mencapai yang mewakili kelas itu.

2.10.1 Euclidean Distance

Dalam menghitung jarak bisa menggunakan konsep *dissimilarity matrix* yaitu dengan mencari nilai perbedaan dari i, j, d_{ij} . Nilai yang diinputkan adalah nilai rata-rata sehingga membentuk suatu nilai yang dapat meminimalkan perbedaan rata-rata setiap entitas (Rainer Hegger, dkk, 2000). Dalam menghitung *dissimilarity* ini bisa menggunakan metode *Euclidean Distance*. Persamaan 2.19 adalah bentuk dari *dissimilarity matrix*.

$$dissimilarities\ matrix = \begin{bmatrix} 0 & \dots & \dots & \dots \\ d(2,1) & 0 & \dots & \dots \\ d(3,1) & d(3,2) & 0 & \dots \\ d(n,1) & d(n,2) & d(n,3) & 0 \end{bmatrix} \quad (2.19)$$

Euclidean Distance adalah ukuran jarak yang paling umum digunakan. Dalam kebanyakan kasus ketika orang mengatakan tentang jarak, mereka akan merujuk ke jarak Euclidean. Jarak euclidean atau hanya "jarak" meneliti akar perbedaan persegi antara koordinat sepasang benda (Gjorgji Madzarov & Dejan Gjorgjevikj, 2010). Di bawah ini merupakan Persamaan 2.20 untuk menghitung jarak euclidean.

$$d_{ij} = \left(\sum (\hat{x}_i - \hat{x}_j)^2 \right)^{\frac{1}{2}} \quad (2.20)$$

Pusat-pusat gravitasi dari dua kelompok yang diperoleh oleh algoritma *clustering* di *node* daun *non* pohon diwakili oleh x_i dan x_j . Persamaan di atas untuk menghitung jarak euclidean, selain dari persamaan di atas kita juga bisa menuliskan rumus untuk menghitung jarak euclidean pada Persamaan 2.21.

$$d_{ij} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2.21)$$

Dua buah titik $p1 = (x1, y1)$ dan $p2 = (x2, y2)$, jaraknya adalah (rumus Euclidean) (Ahmad Shiddiq, dkk, 2012).

2.10.2 Evaluasi Akurasi

Evaluasi digunakan untuk mengetahui tingkat akurasi dari hasil klasifikasi sistem dengan pakar dimana dihitung dengan cara menghitung jumlah data testing yang kelasnya diprediksi oleh sistem. Berikut ini merupakan Persamaan 2.22 untuk prosentase akurasi (Microsoft Translator., 2016).

$$akurasi = \frac{jumlah\ prediksi\ benar}{jumlah\ total\ prediksi} \times 100\% \quad (2.22)$$

Jumlah prediksi benar adalah jumlah prediksi yang sesuai antara prediksi sistem dan prediksi dari pakar sedangkan jumlah total prediksi adalah keseluruhan jumlah data yang diuji.