

BAB II

Pendahuluan

2.1 Tinjauan Pustaka

Pada penelitian sebelumnya, Stikic mencoba untuk melakukan klasifikasi emosi positif dan negatif dengan cara menonton 2 video masing - masing berdurasi 20 menit dan menggunakan EEG sebagai alat untuk mengambil data. Video tersebut terdiri atas video yang dapat mewakili perasaan positif dan negatif. Dengan menggunakan model IDFA, status emosi berhasil diklasifikasikan dengan akurasi sebesar 74,3% sedangkan pada model qDFA akurasi yang diperoleh sebesar 69,4%. Akan tetapi jika menggunakan data perorangan akurasi yang didapatkan bisa mencapai 94,5% (Stikic, et al., 2014).

Lain halnya dengan penelitian yang dilakukan oleh Lu, yang menggunakan kombinasi antara EEG dan *Eye Movement Signals* untuk melakukan klasifikasi status emosi positif, netral dan negatif. Kombinasi tersebut menunjukkan bahwa kombinasi antara EEG dan *Eye Movement Signals* dapat meningkatkan sistem pengenalan emosi. Dengan menggunakan kombinasi antara EEG dan *Eye Movement Signals*, didapatkan akurasi sebesar 87,59% melalui metode *fuzzy integral fusion*. Sedangkan jika data tersebut dipisah, akurasi yang didapatkan hanya sebesar 77,80% dan 78,51% (Lu, et al., 2015).

Penggunaan GSR seringkali untuk mengetahui tingkat stres seseorang, sebagai contoh penelitian yang dilakukan oleh Villarejo. Dia mencoba untuk mengklasifikasi stres seseorang menggunakan GSR dengan cara memberikan induksi stres berupa perhitungan seperti matematika. Hasil yang didapatkan, model yang terbentuk menggunakan data GSR dapat mengklasifikasikan stres

dengan akurasi mencapai 76,56% (Villarejo, et al., 2012).

Kurniawan juga mencoba untuk melakukan hal yang sama dengan Villarejo. Dia mencoba untuk mengklasifikasikan stres dengan menggunakan GSR dan sinyal suara. Pengklasifikasian dengan sinyal suara mendapatkan hasil yang terbaik yaitu sebesar 92% melalui algoritma SVM. Akan tetapi ketika menggunakan data GSR, akurasi yang didapat hanya sebesar 70% melalui algoritma yang sama. Dia juga berpendapat bahwa ketika kita mencoba untuk mengklasifikasikan stres orang lain menggunakan data orang yang lainnya, maka akurasi yang didapatkan akan berkurang (Kurniawan, et al., 2013).

Ketika GSR seringkali digunakan untuk mengklasifikasikan stres seseorang, lain halnya yang dilakukan oleh Liu. Dia mencoba untuk mengklasifikasikan status emosi seperti *happiness, grief, fear, anger* dan *calm* dengan menggunakan data GSR. Tingkat akurasi terbaik yang didapatkan oleh Liu sebesar 66,67% melalui algoritma SVM dengan *kernel radial based function* (RBF) (Liu, et al., 2016).

Belakangan ini, teknik *Sliding Windows* seringkali dipergunakan dalam pengolahan data dalam pengklasifikasian menggunakan data fisiologis. Sebagai contoh, Liu menggunakan data ECG dan Heart rate variability (HRV) yang diolah dengan teknik *Sliding Windows* untuk mengklasifikasikan stres seseorang. Pada penelitian tersebut stres dapat diklasifikasikan dengan akurasi yang tinggi hingga mencapai 98% dengan menggunakan kombinasi data HRV dan ECG yang diolah dengan *Sliding Windows* dengan lag sebesar 24 dan diklasifikasikan menggunakan SVM dengan *kernel linear* (Liu & Ulrich, 2013-2014).

Lain halnya dengan yang dilakukan oleh Wei, yang mencoba untuk mengklasifikasikan emosi seperti *anger*, *fear*, *joy*, *sorrow*, *acceptance*, *rejection*, *surprise* dan *expectancy*. Pada penelitian tersebut, emosi diinduksi menggunakan film dengan pendekatan *color-mood*. Penelitian tersebut mendapatkan hasil yang baik dengan akurasi sebesar 80% yang didapatkan menggunakan algoritma SVM dengan data yang diolah menggunakan teknik *Sliding Windows* (Wei, et al., 2004).

Selain Liu dan Wei, Guo juga mencoba untuk mengklasifikasikan emosi manusia seperti *amusement*, *fear*, *relax* dan *sadness* dengan menggunakan teknik *Sliding Windows*. Akan tetapi Guo menggunakan data yang diambil dengan GSR, diolah menggunakan *Sliding Windows* dengan *lag* sebesar 20. Penelitian tersebut mendapatkan hasil yang baik pula dengan akurasi sebesar 79,45% (Guo, et al., 2013).

2.2 Landasan Teori

2.2.1 Data Preprocessing

Preprocessing seringkali merujuk pada penambahan, pengurangan atau transformasi pada data. Akan tetapi perubahan pada data tersebut bisa menjadi perhatian penting, karena *preprocessing* bisa menciptakan maupun menghancurkan kemampuan model dalam melakukan prediksi. Model yang berbeda juga memiliki tingkat sensitifitas terhadap bentuk data, maka *preprocessing* pada data juga bergantung dari tipe model yang akan digunakan. Sebagai contoh model seperti pohon tidak peka pada karakteristik data, sedangkan *linear regression* peka terhadap karakteristik data (Kuhn & Johnson, 2013).

Sebenarnya, tujuan dari preprocessing data adalah untuk mengubah data *real world* ke dalam suatu format data yang dapat dibaca oleh komputer. Data yang diambil secara langsung dari database atau didapatkan secara *real world* merupakan data yang masih mentah, belum dilakukan transformasi, pembersihan atau perubahan. Sehingga kemungkinan data akan mengandung kesalahan seperti kesalahan input, data yang hilang, dan banyak lainnya (García, et al., 2014).

2.2.2 Penambangan Data

Secara analogi sebutan yang tepat untuk penambangan data adalah penambangan pengetahuan. Akan tetapi banyak orang seringkali menyamakan penambangan data sebagai penemuan pengetahuan, sementara lainnya memandang penambangan data sebagai bagian dari penemuan pengetahuan. Proses pada penemuan pengetahuan terdiri atas beberapa langkah seperti pembersihan data, integrasi data, seleksi data, transformasi data, penambangan data, evaluasi pola dan presentasi pengetahuan.

Fungsi dari penambangan data adalah proses penemuan pola menarik dan pengetahuan dari suatu data berjumlah besar. Data tersebut bisa berasal dari berbagai macam sumber seperti databases, data warehouse, web, dan sebagainya. Secara umum, penambangan data bisa diaplikasikan pada berbagai macam data selama data tersebut memiliki arti pada target yang diinginkan. Ada beberapa fungsionalitas penambangan data termasuk karakterisasi dan diskriminasi, *frequent pattern*, asosiasi, korelasi, *clustering*, klasifikasi dan regresi. Fungsionalitas tersebut berguna untuk menemukan pola yang

diinginkan secara spesifik. Akan tetapi walaupun penambahan data berpotensi untuk dapat menemukan ribuan bahkan jutaan pola atau aturan, tidak semua pola tersebut menarik. Hanya sebagian kecil saja yang dapat menarik perhatian. Pola yang dapat menarik perhatian adalah pola yang mudah dipahami manusia, bisa divalidasi menggunakan data lain dan memiliki potensi yang berguna (Han, et al., 2011).

2.2.3 Supervised Learning

Prediksi sering dianggap sebagai *supervised learning*, karena metode *supervised* mencoba untuk menemukan relasi antara atribut masukan dan atribut target. Hubungan tersebut diwakili pada suatu struktur yang disebut model. Pada umumnya, model mendeskripsikan dan menjelaskan pengalaman, dimana hal tersebut tersembunyi di dalam data dan bisa digunakan untuk memprediksi nilai dari suatu target selama memiliki masukan yang diketahui. *Supervised learning* sering diterapkan pada berbagai macam bidang seperti keuangan, kesehatan, pembangunan dan sebagainya.

Terdapat 2 permasalahan klasik yang termasuk ke dalam *supervised learning*, yaitu klasifikasi dan regresi. Pada klasifikasi atau bisa juga disebut deteksi, atribut target bersifat terbatas dan berupa kategori. Klasifikasi bekerja dengan cara membedakan - bedakan, selama data yang ingin diprediksi mirip seperti data yang dilatih maka kemungkinan hasil prediksi akan tepat. Akan tetapi, nilai yang ingin diprediksi berupa angka pada suatu interval tertentu maka hal tersebut merupakan permasalahan regresi. Namun permasalahan regresi memberikan tingkat kesulitan, kebutuhan komputasi dan kompleksitas

model yang lebih besar dibandingkan permasalahan klasifikasi (García, et al., 2014).

2.2.4 Penambangan Data Temporal

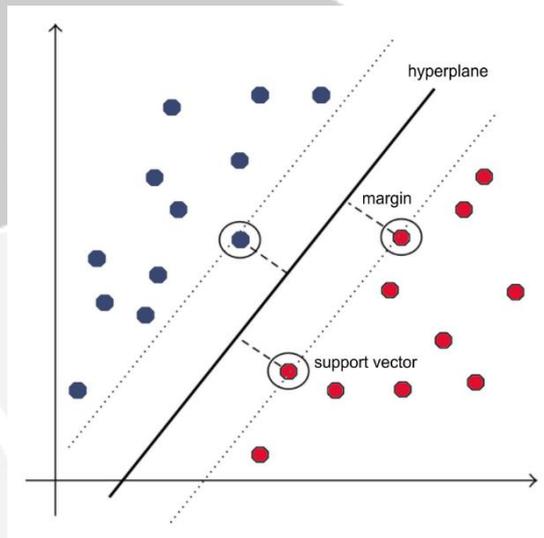
Merupakan teknik penambangan data yang memiliki suatu satuan waktu yang disebut data temporal atau data deret waktu. Hal penting pada penambangan data temporal adalah pencarian suatu kemiripan antara suatu data deret waktu atau deret kejadian. Sebagai contoh kesamaan fluktuasi harga saham dengan data yang lama bisa berguna untuk prediksi harga saham di masa mendatang. Akan tetapi hasil komputasi akan kesamaan data bergantung pada representasi waktu data yang digunakan (Mitsa, 2010).

Antunes berpendapat bahwa tujuan utama dari penambangan data temporal adalah untuk menemukan relasi antara *sequence* dan *subsequence* pada suatu *event*. Dikarenakan data yang bersifat temporal, maka pendekatan yang perlu dilakukan pun juga berbeda. Seringkali, aplikasi yang berhubungan dengan data temporal digunakan untuk melakukan diagnosa dan prediksi perilaku di masa mendatang (Antunes & Oliveira, 2001).

2.2.5 Support Vector Machine

Support Vector Machine (SVM) pertama kali diperkenalkan oleh Vladimir Vapnik pada tahun 1995 yang digunakan untuk keperluan klasifikasi biner. SVM terdiri atas beberapa bagian yang membentuknya seperti *hyperplane* yang merupakan pembatas yang memisahkan setiap kelas dan *support vector* yang merupakan titik yang terdapat pada kedua sisi yang paling mempengaruhi pembentukan *hyperplane*. Konsep dasarnya adalah mencari *hyperplane* yang memisahkan

data d -dimensional secara sempurna menjadi ke dalam 2 kelas. *Hyperplane* yang dicari adalah *hyperplane* yang memiliki margin paling besar, untuk keterangan lebih lanjut bisa melihat Gambar 2.1.



Gambar 2.1. Support Vector Machine

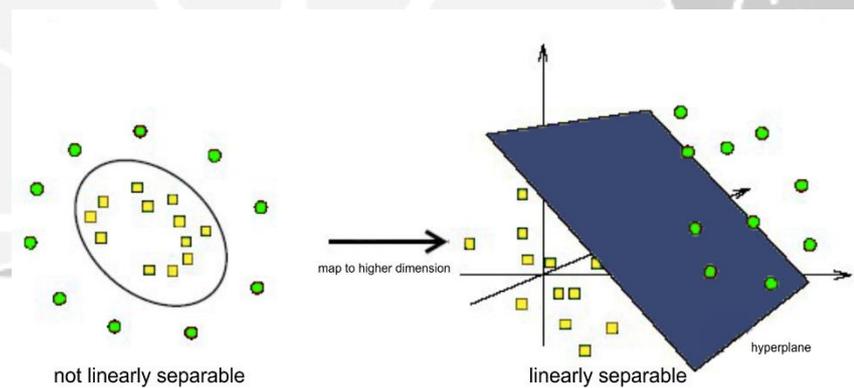
Lalu berikut merupakan rumus untuk menghitung hyperplane pada SVM:

$$y_i(x_i \cdot w + b) \geq 1 \quad \forall_i \dots\dots\dots (2.1)$$

Dimana y_i merupakan *label*, x_i merupakan *input*, w merupakan *weight* dan b adalah *bias*.

Walaupun SVM sudah terbukti untuk menjadi algoritma pembelajaran yang sangat efektif pada data *real world*, tetapi SVM masih belum banyak dipakai. Hal tersebut disebabkan oleh beberapa permasalahan yang terdapat pada SVM seperti kenyataan bahwa implementasi dari SVM yang tidaklah mudah selain itu diperlukan optimasi untuk mendapatkan hasil yang baik. Selain itu penggunaan parameter yang salah dapat meningkatkan tingkat kesulitan pada proses komputasi.

Pada mulanya, SVM diciptakan dengan metode penyelesaian masalah secara linear. Akan tetapi pada masa kini, justru banyak data yang tidak bersifat linear. Maka dari itu, diciptakan *kernels* yang dapat memetakan data ke dalam ruang dimensi yang biasanya lebih besar (*feature space*). Sehingga permasalahan *nonlinear* yang sebelumnya tidak dapat diselesaikan dengan baik dapat terselesaikan. Saat ini terdapat beberapa kernel yang cukup sering digunakan pada SVM seperti *radial basis function* (RBF) atau *gaussian*, *polynomial* dan *sigmoid*. Untuk lebih jelasnya, contoh penyelesaian masalah menggunakan *kernels* dapat dilihat pada Gambar 2.2.



Gambar 2.2. Contoh Penyelesaian Kernels

Akan tetapi, penggunaan *kernels* juga memiliki kelemahan tersendiri. Kesalahan pengeturan parameter pada kernel bisa menyebabkan model menjadi overfit atau underfit. Selain itu, pemakaian kernel juga menambah beban komputasi sehingga menyebabkan proses menjadi lebih lambat.

Salah satu kernel yang sering digunakan adalah RBF, karena kernel tersebut memiliki performa yang baik dan sudah teruji. RBF memetakan data nonlinear ke dalam dimensi yang lebih besar, dengan cara itu

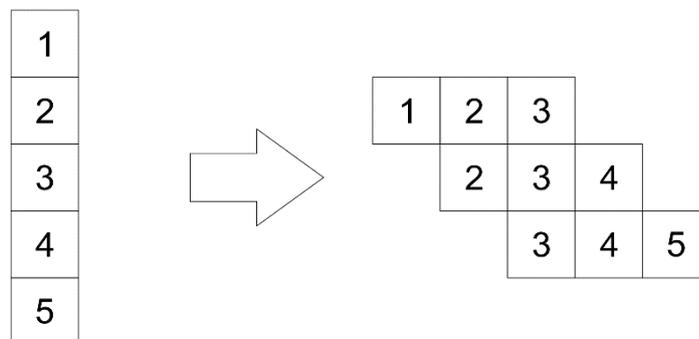
hyperplane pada data nonlinear dapat dicari. Rumus dalam memetakan data pada kernel RBF adalah sebagai berikut:

$$K(x_a, x_b) = \exp\left(-\frac{\|x_a - x_b\|^2}{2\sigma^2}\right) \dots\dots\dots (2.2)$$

Dimana variabel σ bisa diubah sesuai dengan keperluan (Campbell & Cristianini, 1998) (Boswell, 2002) (Campbell & Ying, 2011) (Burges, 1998) (Muller, et al., 2001).

2.2.6 Sliding Windows

Merupakan salah satu metode dalam memproses data temporal yang cukup sering dipakai pada berbagai macam bidang seperti keuangan, biologi dan berbagai permasalahan data temporal lainnya. Metode ini mengubah permasalahan temporal menjadi permasalahan *supervised learning* pada umumnya. *Sliding windows* bekerja dengan cara membentuk *windows classifier* yang dapat memetakan *input window* dengan besaran w untuk suatu nilai *output* y . Secara sederhana, cara kerja metode ini dapat dijelaskan pada Gambar 2.3.



Gambar 2.3. Cara Kerja Sliding Windows

Namun walaupun metode ini sering digunakan untuk memproses data temporal, ada permasalahan utama pada penggunaan *sliding windows* yaitu menentukan ukuran *windows* yang tepat sehingga dapat menghasilkan hasil yang paling optimal. Ada 2 cara yang dapat dilakukan untuk mengatasi permasalahan tersebut, yaitu dengan menggunakan suatu fungsi objektif lalu dibantu menggunakan algoritma optimasi dalam mencari ukuran *window* terbaik dan mencoba untuk menggunakan semua ukuran *window* yang bisa digunakan. Akan tetapi kedua cara tersebut memiliki kelemahan tersendiri, pada penggunaan cara yang pertama ada kemungkinan untuk kehilangan informasi penting yang dapat ditemukan ketika menggunakan *window* dengan ukuran lain. Sedangkan cara kedua tidak memiliki kelemahan ini, namun akan menjadi masalah ketika data yang diolah sangat besar (Dietterich, 2002) (Lijffijt, et al., 2012).

2.2.7 Human-Computer Interaction

Pada tahun 1940an, komputer merupakan barang yang sangat mahal, kompleks dan jarang ditemui. Sehingga hanya ada beberapa orang saja yang berasal dari kalangan pemerintah dan peneliti yang dapat mengoperasikan komputer pada saat itu. Akan tetapi seiring dengan perkembangan jaman, komputer makin sering ditemui di kantor bahkan di rumah. Dengan semakin banyaknya pengguna komputer, menjadikan *Human-Computer Interaction* (HCI) menjadi salah satu topik yang menarik untuk diteliti.

Ilmu yang paling utama dalam HCI adalah *human factors* atau *ergonomics*. *Human factors* yang dimaksud adalah gabungan dari sains dan bidang teknik. Berfokus pada kapasitas manusia, keterbatasan dan

performa dengan desain sistem yang efisien, aman, nyaman dan bahkan menyenangkan bagi manusia yang menggunakan. Dengan mengubah sistem menjadi sistem komputer maka terbentuklah ilmu HCI yang sebenarnya sama dengan *human factors* tetapi dengan fokus hanya pada interaksi manusia dengan teknologi komputasi (MacKenzie, 2013).

2.2.8 Affective Computing

Merupakan proses komputasi yang berhubungan dengan, timbul dari atau dipengaruhi oleh emosi. Pada saat pertama kali, *affective computing* berfokus pada pengenalan, sintesis ekspresi wajah dan sintesis infleksi suara. Namun hal itu hanyalah sebagian kecil saja, karena sebenarnya *affective computing* bisa dilakukan dengan alat pengukuran psikologis yang lainnya. Dengan *affective computing*, maka komputer dapat mengenali dan mengekspresikan emosi yang berguna dalam interaksi dengan manusia.

Dengan keberhasilan dalam mengenali emosi, akan ada banyak aplikasi baru yang dapat diciptakan. Karena emosi berperan penting dalam memahami berbagai macam fenomena yang terjadi seperti perhatian, memori, estetika, pengambilan informasi, hiburan, desain, kesehatan, interaksi manusia dan lain sebagainya dimana *affective computing* bisa diaplikasikan. Selain itu, komputer yang dapat mengenali emosi dapat melakukan pengambilan keputusan dengan lebih pintar (Picard, 1997).

Affective computing yang diperkenalkan oleh Picard pada tahun 1997, terbagi atas empat kategori yaitu:

Tabel 2.1. Empat Kategori *Affective Computing*

Komputer	Tidak bisa mengekspresikan emosi	Bisa mengekspresikan emosi
Tidak bisa mengetahui emosi	I	II
Bisa mengetahui emosi	III	IV

- I. Kebanyakan komputer gagal pada kategori ini, kemampuan komputer dalam mengenali emosi masih lebih rendah dibandingkan hewan pada umumnya. Komputer tersebut tidak bersifat "personal" maupun "user friendly".
- II. Kategori ini berfokus pada pengembangan suara komputer dengan intonasi natural dan "wajah" komputer dengan ekspresi natural. Contohnya adalah aplikasi cortana pada windows. Kategori ini paling maju secara teknologi walaupun masih dalam tahap pengembangan.
- III. Pada kategori ini, komputer bisa mengetahui emosi namun tidak dapat mengekspresikannya. Namun walaupun tidak dapat mengekspresikan emosi, komputer masih dapat memberikan respon.

IV. Pada kategori terakhir, komputer bisa mengetahui dan mengekspresikan emosi. Berpotensi untuk menjadi komputer yang bersifat "personal" dan "user friendly", namun komputer tidak akan terbawa emosi.

2.2.9 Electrodermal Activity

Pertama kali diperkenalkan oleh Johnson dan Lubin pada tahun 1966 sebagai kata umum dalam semua fenomena yang terjadi pada kulit. Merupakan pengukuran efek yang diperantarai secara neurologis pada permeabilitas kelenjar keringat, diamati sebagai perubahan ketahanan kulit terhadap arus listrik kecil atau sebagai perbedaan pada potensial listrik antara bagian kulit yang berbeda. Sinyal Electrodermal Activity merefleksikan aksi dari lalu lintas saraf simpatik pada kelenjar keringat *eccrine*. Aktifitas saraf simpatik berhubungan erat dengan perubahan keadaan mental seseorang.

EDA merupakan parameter yang sangat berguna karena mudah diukur dan handal. Akan tetapi EDA sensitif terhadap berbagai macam ransangan, maka diperlukan interpretasi yang cermat. Sampai saat ini, EDA telah banyak digunakan sebagai indeks *physiological* dan *emotional arousal* dalam penelitian pada responsivitas stres, termasuk percobaan dalam kejutan rasa sakit, film emosional dan ketahanan mental dan fisik (Boucsein, 2012) (Gellman & Turner, 2013).

EDA juga sering disebut sebagai skin conductance (SC) dan galvanic skin response (GSR). Menurut Ayata, dengan GSR berbagai macam dampak dari hal yang membangkitkan emosi bisa di tes dengan menggunakan

gambar, suara, video, kegiatan, dan sebagainya. GSR menghasilkan sinyal psikologis yang merupakan refleksi dari ketertarikan manusia terhadap suatu hal, ketika manusia tertarik maka akan mengeluarkan keringat sehingga meningkatkan kadar garam dan menambah hambatan listrik di kulit (iMotions, 2016) (AYATA, et al., 2017).

2.2.10 R Language

R merupakan bahasa dan lingkungan open source untuk perhitungan statistik dan grafik, mirip seperti S Language yang dikembangkan oleh Bell Labs. Hampir setiap kegiatan analisis data dapat dilakukan dengan R. Tingkat perkembangan package yang pesat membuat R menjadi salah satu platform yang kuat dalam analisis data.

Dalam melakukan klasifikasi, R menyediakan banyak package yang dapat menunjang proses klasifikasi. Walaupun package tersebut sudah disediakan, bukan berarti cara kerjanya tidak dapat diketahui. Karena R menyimpan semua log syntax yang telah digunakan, sehingga semua fungsi yang sudah digunakan dapat dilihat dan dipelajari cara kerjanya (Kabacoff, 2015).