

APLIKASI MOBILE (LIDE) UNTUK DIAGNOSIS TINGKAT RESIKO PENYAKIT STROKE MENGGUNAKAN PTVPSO-SVM

Muhammad Syafiq¹, Achmad Jafar Al Kadafi², Rizka Husnun Zakkiyah³, Daeswara Jauhari⁴, Wanda Athira Luqyana⁵, Imam Cholissodin⁶, Lailil Muflikhah⁷

^{1,2,3,4,5,6,7} Fakultas Ilmu Komputer, Universitas Brawijaya

Email: ¹syafiq.rezpector@gmail.com, ²al.kadafi20@gmail.com, ³be.scientist.in.future@gmail.com,

⁴daneswarajauhari@gmail.com, ⁵wandathira@gmail.com, ⁶imamcs@ub.ac.id, ⁷lailil@ub.ac.id

(Naskah masuk: 20 Mei 2016, diterima untuk diterbitkan: 20 Juni 2016)

Abstrak

Penyakit Stroke merupakan penyakit yang umum dan menduduki peringkat kedua dalam kematian di Indonesia dengan angka 11,13%. Penyakit yang menyerang fungsi saraf otak dengan tingkat resiko bertahap. Dengan tidak hanya menyerang pada manusia usia lanjut, kini penyakit stroke dapat menyerang siapa saja. Indikator yang dapat digunakan dalam mewaspadai tingkat stroke adalah memeriksa kolesterol tubuh, HDL, LDL, dan Triglisierida. Namun faktor umum juga menjadi masalah utama namun umum. Dengan bertambahnya umur 10 tahun dimulai dari umur 55 tahun, maka tingkat resiko penyakit stroke menjadi dua kali lipat. Tingkat kewaspadaan perlu diperhatikan, sehingga dalam sewaktu-waktu untuk memeriksakan kondisi tubuh diperlukan. Lide adalah sebuah aplikasi pada smarthphone yang berbasis Android yang mengimplementasikan perhitungan algoirtma dalam menentukan tingkat resiko penyakit stroke. Lide merupakan salah satu solusi yang dibangun dalam mengontrol tingkat resiko stroke, hanya dengan memasukkan tingkat kolesterol, LDL, HDL, dan triglieserida. Dilengkapi dengan treatment bagi para pengguna, sehingga dapat melakukan penanganan awal pada tingkat-tingkat tertentu. Dalam aplikasi bergerak ini mengimplementasikan metode Particle Swarm Optimization dan Support Vector Machine untuk klasifikasi dengan menggunakan 4 fitur (kolesterol, HDL, LDL, dan Triglisierida). Dengan menggunakan metode ini, nilai akurasi yang dihasilkan mencapai 87%.

Kata kunci: Stroke, PTVPSO-SVM, Aplikasi Perangkat Bergerak, Lide

Abstract

Stroke is a common disease and was ranked second in the number of death in Indonesia with 11, 13%. This disease attacks the nerve function of brain with the level of risk. Not only attacks on old age, this disease can attacks everyone. Indicators that can be used in reasonably level stroke is cholesterol, HDL, LDL, and Triglycerides. But the common factor is also becoming a major problem, however. Increasing public 10 years starting at age 55 years, then the risk level of stroke become twice. The level of alertness to note, so in any time to check the condition of the body is required. Lide is an application on an android-based smartphone which implement algorithm calculation to determining the risk level of stroke. Lide is one of solution that is built to control the risk level of stroke, by simply entering the level of LDL, HDL, Cholesterol, and Triglycerides. It is equipped with treatment for users, so that it can perform the initial handling at certain levels. This mobile application implements the method of Particle Swarm Optimization and Supprot Vector Machine for classification by using 4 features (cholesterol, LDL, HDL, and Triglycerides). By using this method, the value of the resulting accuracy is reached 87%.

Keywords: Stroke, PTVPSO-SVM, Mobile Apps, Lide

1. PENDAHULUAN

Indonesia disebut sebagai negara agraris, hal ini Stroke merupakan penyakit yang memiliki nama ilmiah *Cerebrovascular Accident* (CVA) yang menyerang fungsi saraf otak dengan intensitas peningkatan penyakit yang beragam, dapat dikategorikan penyakit stroke ringan hingga berat dan tingkat pemulihan yang secara cepat maupun perlahan. Terjadinya stroke dikarenakan adanya penyumbatan dalam aliran darah pada otak sehingga dapat mengakibatkan gangguan pada system motorik (kelumpuhan). Stroke merupakan penyakit penyebab kematian dengan peringkat kedua dengan angk

11,13% dari keseluruhan total kematian diseluruh dunia (Mozaffarian, 2015). Berdasarkan data riset di Indonesia pada tahun 2013, kasus stroke mencapai 12,1 dari 1000 orang dan perkembangan penyakit dapat terus meningkat dengan adanya factor-faktor resiko dan penduduk usia lanjut (Kementrian, 2013).

Dengan peningkatan penderita stroke yang cukup tinggi pada setiap tahun, belum terdapat upaya efektif dalam menanggulangi penyakit baik dengan meningkatkan kesadaran masyarakat maupun pengelolaan penyakit stroke yang optimal. Tingkat kesadaran terhadap bahaya stroke yang kurang dapat merugikan masyarakat karena penanganan yang terlambat sehingga penderita dapat mengalami kelumpuhan bahkan kematian. Dalam deteksi stroke

di Indonesia telah diterapkan dua cara, yaitu dengan *Computed Tomography (CT) Scan* dan *Magnetic Resonance Imaging (MRI)*. Namun dengan menggunakan pemeriksaan CT Scan maupun MRI cukup menguras biaya sekitar 1 – 4 juta rupiah, sehingga tidak seluruh orang dapat menjalani deteksi stroke secara dini.

Klasifikasi terkait dengan penyakit stroke telah dilakukan dengan berbagai metode. Metode Support Vector Machine (SVM) dan regresi logistik digunakan dalam klasifikasi kondisi pasien stroke setelah pengobatan. Penelitian tersebut dilakukan oleh Dwi Ratnasari dengan melibatkan 3 variabel yaitu usia, *Transient Ischemic Attack (TIA)*. Dengan kedua metode tersebut dihasilkan prosentase SVM 80% dan regresi logistik 74,19% (Ratnasari, 2013).

SVM merupakan metode yang menghasilkan data yang akurat namun dengan terdapat kendala saat pemilihan parameter yang kurang tepat. Dalam mengatasi parameter yang kurang tepat adalah meningkatkan akurasi dengan seleksi nilai terbaik dari setiap parameter. Algoritma *Particle Swarm Optimization (PSO)* dan Algoritma Genetika merupakan contoh algoritma yang berguna untuk mengoptimasi setiap parameter. Berdasarkan hasil penelitian yang dilakukan oleh Yuniar Marbun, PSO menghasilkan data yang lebih stabil dalam optimasi penjadwalan matakuliah (Marbun, 2013).

Berdasarkan fakta dan permasalahan yang ada dalam klasifikasi penyakit stroke, maka penulis mengajukan rekayasa perangkat lunak “*Lide*” dengan menggunakan metode SVM dan PSO. Rekayasa perangkat lunak diharapkan dapat mempermudah pengguna dalam deteksi stroke secara dini.

2. STUDI PUSTAKA

2.1 Stroke

Penyakit stroke didefinisikan sebagai terjadinya gangguan fungsional otak fokal maupun global secara mendadak yang dikarenakan gangguan aliran darah pada otak yang berlangsung lebih dari 24 jam. Gangguan aliran darah pada otak sebagai penyuplai darah yang membawa oksigen dan glukosa yang diperlukan untuk metabolisme sel-sel otak agar dapat melaksanakan fungsinya pun terhambat. Terdapat beberapa penyebab terjadinya stroke, diantaranya adalah penyumbatan kolestrol. (WHO MONICA, 1990).

Tingkat keparahan stroke dapat berkembang dengan cepat, baik dari tingkat keparahan ringan hingga berat. Namun stroke dapat terjadi dengan tingkat keparahan yang stabil bahkan dapat membaik. Dalam penyakit stroke, tidak mengenal usia muda maupun tua, tetapi pada umumnya penyakit tersebut diderita oleh manula dengan usia diatas 55 tahun. Cara hidup sehat merupakan salah satu pencegahan untuk menghindari stroke. Pola makan yang berlebihan baik yang mengandung banyak garam

maupun lemak, mengurangi rokok, mengurangi untuk berfikir berat (stres), dan olahraga.

2.2.1 Jenis-Jenis Stroke

Gangguan peredaran darah di otak dapat menyebabkan stroke. Namun penyakit stroke masih digolongkan menjadi dua jenis yaitu stroke iskemik, yang disebabkan karena penyumbatan pembuluh darah otak dan stroke hemoragik disebabkan oleh pecahnya pembuluh darah otak. (Yastroki, 2012)

1. Stroke Iskemik

Proses stroke yang disebabkan oleh penyumbatan pembuluh darah otak yang menyebabkan berkurangnya suplai oksigen dan glukosa ke otak. Jaringan otak akan mati jika suplai oksigen secara berkelanjutan terus berkurang, sehingga pembekuan darah dapat terjadi.

2. Stroke Hemoragik

Selain karena penyumbatan, pendarahan didalam otak dapat terjadi sehingga mengganggu aliran darah ke otak. Stroke hemoragik menjadi lebih berbahaya dan menyebabkan kematian. Pendarahan yang terjadi pada stroke hemoragik dibagi menjadi dua jenis dengan pendarahan yang terjadi didalam otak dan pendarahan diantara bagian dalam dan luar lapisan jaringan yang melindungi otak.

2.1.1 Faktor Risiko Stroke

Stroke merupakan penyakit yang perlu dihindari dan harus cepat untuk dicegah. Pencegahan dapat dimulai dari mengetahui faktor-faktor penyebab stroke yang telah menjadi penyakit dengan angka kematian tinggi, nomor tiga di Indonesia (Utami, 2009). Faktor stroke dapat timbul dari berbagai penyebab, sehingga dikenal dengan multikausal dalam dunia kedokteran.

Faktor-faktor penyebab stroke telah dibagi menjadi beberapa kelompok oleh beberapa ahli. Menurut Sutrisno, faktor stroke dibagi menjadi dua kelompok yaitu, faktor yang tidak dapat diubah yang dikarenakan: keturunan, jenis kelamin, umur, dan ras. Dan faktor yang dapat diubah adalah obesitas, diabetes mellitus, penyakit jantung dan hipertensi, hiperkolesterol, serta gaya hidup (Sutrisno, 2007). Sedangkan menurut Yastroki, stroke dapat dibedakan menjadi tiga kelompok yaitu, faktor resiko yang tidak dapat dimodifikasi, faktor perilaku (primordial), dan faktor sosial ekonomi (Yastroki, 2013). Namun faktor-faktor umum yang perlu diperhatikan dalam menangani stroke setelah pemeriksaan laboratorium adalah:

1. Umur

Pada umumnya penyakit stroke sering diderita oleh lansia dengan rata-rata umur diatas 55 tahun, namun faktanya penyakit stroke tidak mengenal usia. Peningkatan penyakit stroke dapat terjadi dua kali lipat setiap penambahan umur 10 tahun dimulai dari umur 55 tahun (Soeharto, 2004).

2. Total Kolesterol

Pengelompokan total kolesterol dimulai dari normal jika rentan nilainya dibawah 200 mg/dl. Untuk kolesterol tinggi dengan rentan nilai 200-239 mg/dl. Dan tingkat kolesterol yang sangat tinggi adalah nilai yang lebih dari 239 mg/dl (Soeharto, 2004).

3. HDL (High Density Lipoprotein)

Terdapat tiga kategori untuk HDL yaitu, dikatakan rendah jika nilai HDL kurang dari 35 mg/dl. HDL dapat menguntungkan tubuh jika rentan nilainya dari 35-60 mg/dl, dan dikategorikan sangat menguntungkan jika nilai HDL lebih dari 60 mg/dl.

4. LDL (Low Density Lipoprotein)

LDL merupakan kolestrol jahat karena peredaran kolestrol yang menuju sel-sel jaringan liver. LDL dikategorikan menjadi tiga kelompok dengan nilai kurang dari 100 mg/dl, LDL dikatakan normal. Jika nilai LDL 100-199 mg/dl telah dikategorikan sebagai LDL batas tinggi. Dan dengan nilai lebih dari 199 mg/dl maka LDL telah dikategorikan sangat tinggi, sehingga membahayakan tubuh.

5. Trigliserida

Kelebihan kalori pada tubuh yang dapat dikatakan sebagai lemak dalam darah dapat berfungsi untuk tubuh sebagai sumber energi pengganti. Trigliserida dikatakan normal jika nilainya kurang dari 150 mg/dl. Kategori batas tinggi jika rentan nilainya dari 150 – 199 mg/dl. Kategori tinggi jika nilai trigliserida adalah 200 – 499 mg/dl. Dan dikategorikan sangat tinggi sehingga tidak baik untuk tubuh jika nilainya lebih dari 500 mg/dl.

2.2 Klasifikasi

Klasifikasi adalah proses dari pengelompokan data maupun fakta-fakta tertentu yang telah memenuhi suatu kriteria. Terdapat dua kelompok klasifikasi yang dibedakan yaitu, klasifikasi sederhana dan klasifikasi kompleks. Yang dimaksud dengan klasifikasi sederhana ialah bentuk klasifikasi yang mengelompokkan objek hanya dengan dua kelompok dan klasifikasi kompleks adalah klasifikasi yang mengelompokkan objek menjadi tiga kelompok atau lebih. Namun deskripsi lain tentang klasifikasi dikemukakan oleh Han dan Amber yang mengartikan klasifikasi adalah model yang terdapat pada data mining dimana classifier dikonstruksi untuk melakukan prediksi kategori atau kelas dari suatu data (Han, 2006).

Dalam proses klasifikasi harus dapat memenuhi kriteria yang logis dan konsisten. Sehingga dalam penentuan kelompok klasifikasi harus diperhatikan kriteria-kriteria yang memenuhi sesuai dengan masalah yang ada (Pramudiono, 2006).

Learning dan test merupakan fase yang harus dilalui saat penentuan klasifikasi. Dengan fase learning merupakan proses untuk membentuk model

prediksi pada sebagian data yang telah diketahui kelas datanya. Model-model yang telah terbentuk, kemudian diuji dengan data lainnya untuk mendapatkan akurasi pada model, hal ini dilakukan pada fase test. Hasil dari akurasi model yang mencukupi, dapat digunakan sebagai pedoman untuk memprediksi kelas data lain yang belum diketahui (Pramudiono, 2006).

2.4 Support Vector Machine (SVM)

Support Vector Machine adalah salah satu metode untuk menyelesaikan pengenalan pola yang diperkenalkan pada tahun 1992 oleh Vapnik, Bose, dan Guyon. Support Vector Machine (SVM) adalah sistem pembelajaran yang menggunakan ruang hipotesis berupa fungsi-fungsi linier dalam sebuah ruang fitur berdimensi tinggi.

Metode Support Vector Machine digunakan untuk melakukan proses klasifikasi dari data yang telah memiliki fitur-fitur yang nilainya sudah ditentukan. Untuk melakukan klasifikasi dengan metode ini, perlukan data training untuk melakukan pembelajaran atau learning serta data uji melakukan proses klasifikasinya. Proses perhitungan klasifikasi data dapat dilalui dengan beberapa proses, yaitu:

1. Menentukan nilai parameter *augmenting factor* (λ), *learning rate* (α), serta konstanta (C).
2. Menghitung Kernel

$$K(x, y) = \exp \left(-\frac{\|x-y\|^2}{2\sigma^2} \right) \quad (1)$$

3. Perhitungan One Against All
4. Melakukan Sequential Learning yaitu yang menjalankan proses iterasi pelatihan α sampai Penentuan nilai α_i . Dengan tahapan proses sequential learning sebagai berikut :

- a. Menghitung Matriks $[D]_{ij}$

$$[D]_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (2)$$

- b. Menghitung nilai error E_i

$$E_i = \sum_{j=1}^n \alpha_j D_{ij} \quad (3)$$

- c. Menghitung nilai pembatas Lagrange Multiplier $\delta \alpha_i$

$$\delta \alpha_i = \min(\max(\gamma(1-E_i), -\alpha_i), C-\alpha_i) \quad (4)$$

- d. Menghitung multiplier α_i baru.

$$\alpha_i = \alpha_i + \delta \alpha_i \quad (5)$$

5. Menghitung nilai bias b

$$b = -\frac{1}{2} \sum_{i=1}^n \alpha_i y_i (K(x_i, x^+) + K(x_i, x^-)) \quad (6)$$

6. Menghitung nilai fungsi $f(x)$

$$f(x) = \text{sign}(\sum_{i=1}^n \alpha_i \cdot y_i \cdot K(x, x_i) + b) \quad (7)$$

7. Perhitungan Akurasi

$$\text{Akurasi} = \frac{TP+TN}{TP+TN+FP+FN} \cdot 100\% \quad (8)$$

2.5 Particle Swarm Optimization (PSO)

Teknik optimasi dengan cara menghitung secara terus menerus calon solusi dengan menggunakan suatu acuan kualitas. Algoritma ini mengoptimasi permasalahan menggunakan fungsi tertentu untuk posisi dan kecepatan dari partikel. Sekumpulan partikel ini dinamakan swarm, dan pada akhirnya swarm akan bergerak menuju kepada solusi terbaik (Chen, 2011) dan (Soebroto, 2015). Proses perhitungan *Parallel Time Variant* (PTVPSO) dilalui oleh beberapa tahap sebagai berikut:

1. Inisialisasi posisi dan velocity awal

$$X_0^i = X_{min} + rand(X_{max} - X_{min}) \quad (9)$$

$$V_0^i = X_{min} + rand(X_{max} - X_{min}) \quad (10)$$

2. Proses update velocity dan posisi

$$V_{k+1}^i = w * V_k^i + c1 + rnd * (p^i - X_k^i) + c2 + rnd * (p_k^g - X_k^i) \quad (11)$$

$$w = w_{min} + (w_{max} - w_{min}) \frac{(t_{max} - t)}{t_{max}} \quad (12)$$

$$c_1 = c_{1i} + (c_{1f} - c_{1i}) \frac{t}{t_{max}} \quad (13)$$

$$c_2 = c_{2i} + (c_{2f} - c_{2i}) \frac{t}{t_{max}} \quad (14)$$

Proses update posisi dari velocity :

$$X_{k+1}^i = X_k^i + V_{k+1}^i \quad (15)$$

3. METODOLOGI

3.1. Data

Pada penelitian ini data yang digunakan adalah data yang dibutuhkan pada proses klasifikasi tingkat resiko penyakit stroke. Data yang digunakan dibagi menjadi dua bagian. Pertama data yang digunakan untuk proses *learning* pada sistem, kedua merupakan data yang digunakan untuk proses testing.

3.2. Tahapan Penelitian

Pembuatan aplikasi Lide, melalui dari beberapa tahapan proses. Tahapan proses yang telah dilakukan adalah:

1. Desain Tampilan
User Interface dibuat agar memudahkan pengguna dalam berinteraksi dengan Lide *Mobile Apps*.
2. Implementasi Algoritma SVM
Diterapkan algoritma SVM dan PSO dalam perhitungan untuk mendapatkan hasil akhir tingkat resiko penyakit stroke.
3. Evaluasi
Evaluasi merupakan tahapan dengan menghitung akurasi dari hasil yang didapat pada implementasi algoritma.

3.3. Perhitungan Manual

Perhitungan manual menggunakan SVM menggunakan 15 data latih, dan 12 data uji untuk mengklasifikasikan 3 tingkat resiko penyakit stroke. Adapun tahapannya sebagai berikut.

1. Menentukan Nilai Parameter

Beberapa parameter seperti *augmenting factor* (λ), *learning rate* (α), serta konstanta biaya (C) harus ditentukan terlebih dahulu sebelum metode SVM dapat dijalankan. Parameter tersebut ditentukan secara acak. Pada Tabel 3.1 berikut adalah nilai dari parameter yang dibutuhkan.

Table 3.1 Nilai Parameter SVM

λ	γ	C
0,1	0,2	1000

2. Menghitung Kernel

Fungsi kernel yang digunakan para penelitian ini adalah Gaussian RBF dengan rumus sebagai berikut.

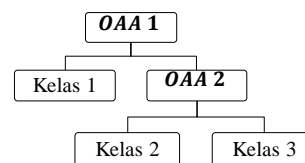
$$K(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \quad (16)$$

Pada penelitian ini terdapat 5 parameter yang digunakan sehingga digunakan fungsi *tunneling* untuk menghitung fungsi kernel.

$$K(l_1, l_2) = \prod_{p=1}^5 \exp\left(-\frac{\|l_1^p - l_2^p\|^2}{2\sigma_p^2}\right) \quad (17)$$

3. Penentuan *One Against All*

Klasifikasi dilakukan dengan cara pembagian kelas positif dan kelas negatif sebanyak $(n-1)$ kali dimana n merupakan jumlah kelas yang ada. Pada contoh terdapat 3 kelas klasifikasi sehingga akan dilakukan pembagian kelas positif dan kelas negatif sebanyak 2 kali. Gambar 3.1 adalah contoh skema penentuan one against all (OAA).



Gambar 3.1 Skema OAA

Pada Tabel 3.2 adalah hasil penentuan *One Against All* pada contoh data latih

Table 3.2 Hasil OAA Data Latih

No	Status Resiko	y_1	y_2
1	Tinggi	+	-
2	Tinggi	+	-
⋮	⋮	⋮	⋮
6	Sedang	-	+
⋮	⋮	⋮	⋮
15	Rendah	-	-

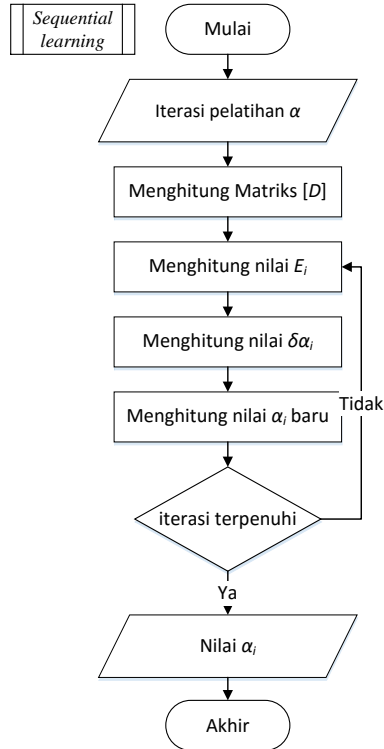
Keterangan

+: Kelas Positif

-: Kelas Negatif

4. Perhitungan *Sequential Learning*

Proses *Sequential Learning* digunakan untuk melatih nilai *multiplier* (α) dari data latih yang ada. Secara umum proses *sequential learning* ditunjukkan pada Gambar 3.2 berikut.



Gambar 3.2 Proses Sequential Learning

4.1 Menghitung Matrix D

Pada perhitungan matrix D membutuhkan nilai *kernel* masing-masing data dan nilai *augmenting factor* (λ). Berikut adalah contoh perhitungan matriks [D] data latih 1 terhadap data latih 2 untuk *One Against All* level 1.

$$[D]_{12} = y_1 y_2 (K(l_1, l_2) + \lambda^2) \quad (18)$$

$$[D]_{12} = 1 \cdot 1 (1 + 0.1^2)$$

$$[D]_{12} = 1.01$$

Pada Tabel 3.3 adalah contoh perhitungan matrik D pada data latih untuk *One Against All* tingkat 1.

Table 3.3 Hasil Matrik D

Data	1	2	...	15
1	1.01	0.010939	...	-0.11918
2	0.0109392	1.01	...	0.009997
3	0.0107948	0.033321	...	0.009779
4	0.0135727	0.014275	...	0.008441
5	0.0100002	0.010628	...	0.01

6	-0.071315	-0.02218	...	0.02564
...
15	-0.119183	0.009997	...	1.01

4.2 Menghitung nilai error E

Pada perhitungan E_i digunakan untuk menghitung nilai error berdasarkan matrix D dan *multiplier* α . Proses ini membutuhkan nilai matrix D dan nilai *multiplier* $\alpha_{(iterasi-1)}$. Nilai inisialisasi *multiplier* α_0 diisi sams dengan 0. Berikut adalah contoh perhitungan E_1 pada iterasi 1 untuk saat *One Against All* level 1.

$$E_1 = \sum_{j=1}^{15} \alpha_j D_{1j} \quad (19)$$

$$= (0 \cdot 1.01) + (0 \cdot 0.0109) + \dots + (0 - 0.119)$$

$$= 0$$

Pada Tabel 3.4 adalah contoh perhitungan E_i pada data latih untuk *One Against All* tingkat 1.

Table 3.4 Hasil E_i

E	Iterasi 1
1	0
2	0
3	0
...	...
15	0

4.3 Menghitung pembatas *Lagrange Multiplier* $\delta\alpha_i$

Proses ini digunakan untuk membatasi agar nilai *multiplier* α yang baru selalu bernilai positif ($\alpha \geq 0$) dan selalu kurang dari konstanta yang telah ditetapkan ($\alpha \leq C$). Berikut adalah contoh perhitungan $\delta\alpha_i$ pada data latih 1, iterasi 1, untuk *One Against All* level 1.

$$\delta\alpha_1 = \min(\max(\gamma(1 - E_1), -\alpha_1), C - \alpha_1) \quad (20)$$

$$= \min(\max(0.2 \cdot (1 - 0), -0), 1.000 - 0)$$

$$= \min(\max(0.2, 0), 1.000)$$

$$= \min(0.2, 1.000)$$

$$= 0.2$$

Pada Tabel 3.5 adalah contoh hasil dari perhitungan $\delta\alpha_i$ pada iterasi 1, untuk *One Against All* level 1.

Table 3.5 Hasil $\delta\alpha_i$

$\delta\alpha_i$	Iterasi 1
1	0.2
2	0.2
3	0.2
...	...
14	0.2
15	0.2

4.4 Menghitung *multiplier* baru α_i

Proses ini digunakan untuk memperbarui *multiplier* seiring bertambahnya iterasi dalam *learning*. Berikut adalah contoh perhitungan *multiplier* α_i baru pada data 1, iterasi 1, untuk *One Against All* level 1.

$$\alpha_i = \alpha_i + \delta\alpha_i \quad (21)$$

$$= 0 + 0,2$$

$$= 0,2$$

Pada Tabel 3.6 adalah contoh hasil dari *multiplier* α_i baru pada iterasi 1, untuk *One Against All* level 1

Table 3.6 Hasil α_i

Data	α lama	α baru
1	0	0.2
2	0	0.2
3	0	0.2
4	0	0.2
5	0	0.2
⋮	⋮	⋮
15	0	0.2

Pada Tabel 3.7 adalah nilai dari *multiplier* α_i untuk semua iterasi pada saat *One Against All* level 1.

Table 3.7 Hasil α_i Semua Iterasi

Data	α_0	α_1	α_2	α_3
1	0	0.2	0.382772	0.5404092
2	0	0.2	0.362324	0.4927641
3	0	0.2	0.370588	0.5123627
4	0	0.2	0.393791	0.5648823
5	0	0.2	0.356492	0.4775039
⋮	⋮	⋮	⋮	⋮
15	0	0.2	0.314762	0.3878847

5. Menghitung bias

Sebelum nilai bias dapat dihitung, ditentukan terlebih dahulu data yang memiliki nilai *multiplier* terbesar untuk kelas positif dan negatif. Data latih yang memiliki nilai *multiplier* α_i terbesar untuk kelas positif dinotasikan dengan (x^+) , sedangkan kelas negatif dinotasikan dengan (x^-) . Berikut adalah contoh perhitungan *bias* untuk *One Against All* level 1.

$$b = -\frac{1}{2} \sum_{i=1}^{15} \alpha_i y_i (K(x_i, x^+) + K(x_i, x^-)) \quad (22)$$

$$b = -\frac{1}{2} \left(\begin{array}{c} (0.54 \cdot 1 \cdot (0.003 + 0.00004)) \\ + \dots + \\ (0.38 \cdot -1 \cdot (0.001 + 0.0001)) \end{array} \right)$$

$$b = -\frac{1}{2} (-0.18913349)$$

$$b = 0.094566745$$

6. Menghitung fungsi $f(x)$

Fungsi $f(x)$ digunakan untuk mengetahui *hyperplane* dari permasalahan berdasarkan support vector yang ada. Pada proses ini membutuhkan nilai bias, *multiplier* terbaru, nilai kelas dari data latih yang bersangkutan, serta nilai kernel dari data uji dengan masing-masing data latih. Sebagai contoh terdapat data uji dengan spesifikasi seperti pada Tabel 3.8.

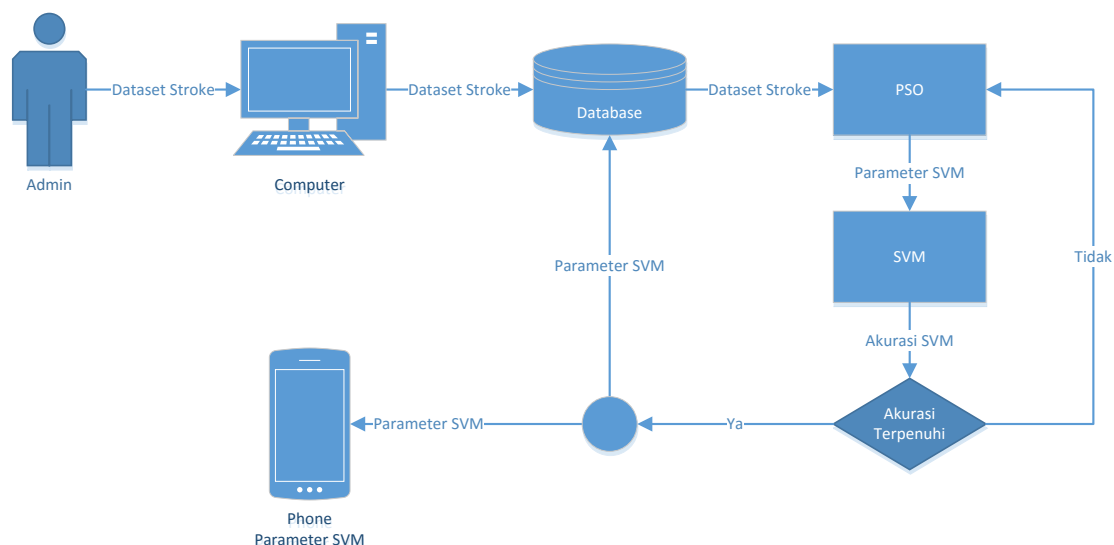
Table 3.8 Parameter dan Status Resiko

No	Parameter					Status Resiko	
	P1	P2	P3	P4	P5	Sistem	Dokter
1	160	192	285	183	63		Tinggi

Maka perhitungan manual untuk menghitung fungsi $f(x)$ adalah sebagai berikut.

$$f(u_1)_{level_1} = \text{sign}(\sum_{i=1}^{15} \alpha_i \cdot y_i \cdot K(u_1, l_i) + b) \quad (23)$$

$$= \text{sign} \left(\left(\begin{array}{c} (\alpha_1 \cdot y_1 \cdot K(u_1, l_1)) \\ + \dots + \\ (\alpha_{15} \cdot y_{15} \cdot K(u_1, l_{15})) \end{array} \right) + b \right)$$



Gambar 3.2 Proses Aplikasi Lide

$$= \text{sign} \left(\left(\frac{(0.5404 \cdot 1 \cdot 0.0001)}{(\dots \cdot \dots \cdot \dots)} + 0.0945 \right) \right)$$

$$= \text{sign}(0.0200)$$

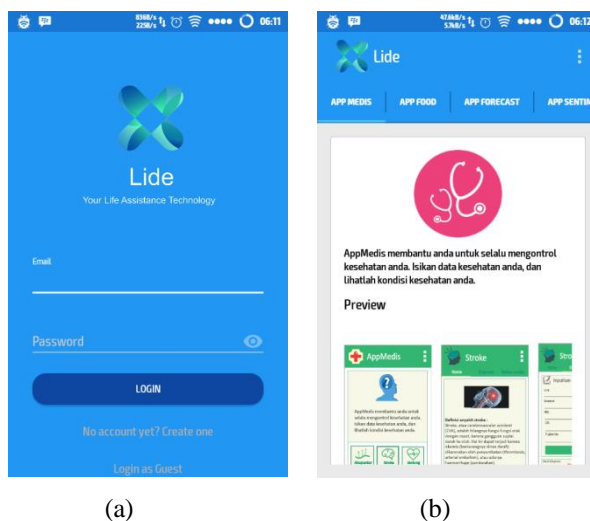
$$= 1$$

Nilai perhitungan untuk data uji pertama menghasilkan nilai 1 pada fungsi $f(x)$ level 1 sehingga data uji tersebut diklasifikasikan kepada kelas yang bernilai 1 atau (+) untuk level 1 pada penentuan *One Against All* yaitu kelas “Tinggi”. Apabila nilai yang dihasilkan pada perhitungan $f(x)$ level 1 adalah -1. Maka data uji tersebut harus dilakukan perhitungan $f(x)$ kembali, tetapi parameter yang digunakan adalah $f(x)$ level 2.

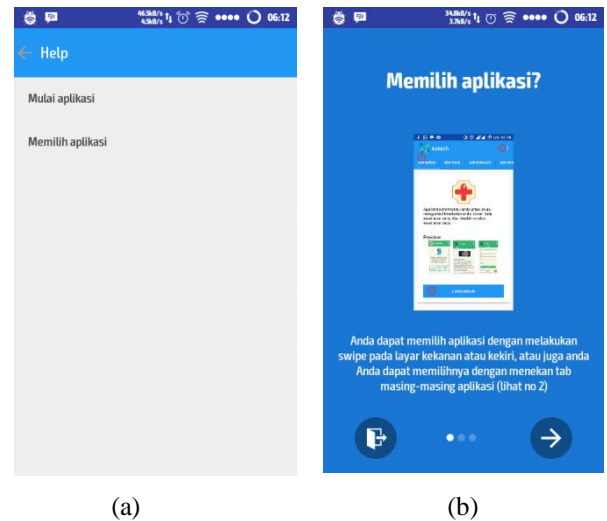
4. HASIL DAN PEMBAHASAN

Pada proses aplikasi Lide dapat digunakan pada smartphone dengan spesifikasi seperti pada Gambar 3.3. Proses dalam menjalankan aplikasi ini yaitu dengan perhitungan dari nilai-nilai parameter yang telah dimasukkan oleh pengguna yang berupa LDL, HDL, Kolesterol, dan Trigliserida. Perhitungan untuk menentukan tingkat resiko stroke digunakan metode *Particle Swarm Optimization* (PSO) dan *Support Vector Machine* (SVM).

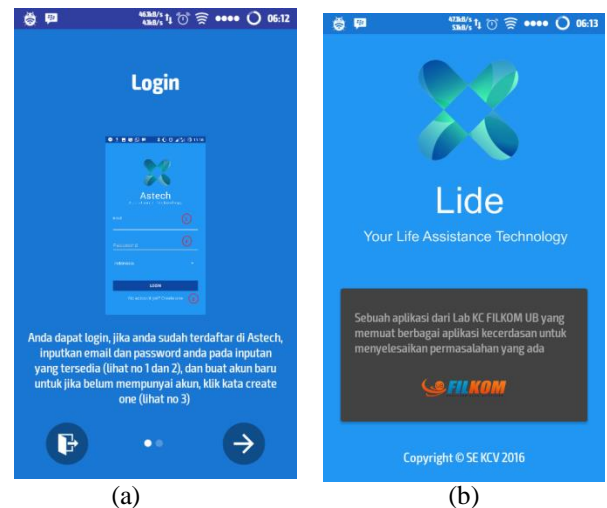
Aplikasi Lide terdiri dari beberapa tampilan, yang didesain sederhana dan sedemikian rupa, sehingga memudahkan pengguna untuk instalasi dan juga ketika menggunakan aplikasi tersebut termasuk dari pilihan warna, gambar yang digunakan dan berbagai fitur-fitur yang ada di dalamnya. Antarmukanya terdiri dari Halaman Awal, Main Apps, Menu Help, Login, About, App Medis, App Stroke, Diagnosis, Treatment, Medical Record, dan Chart seperti pada Gambar 4.1 – 4.7.



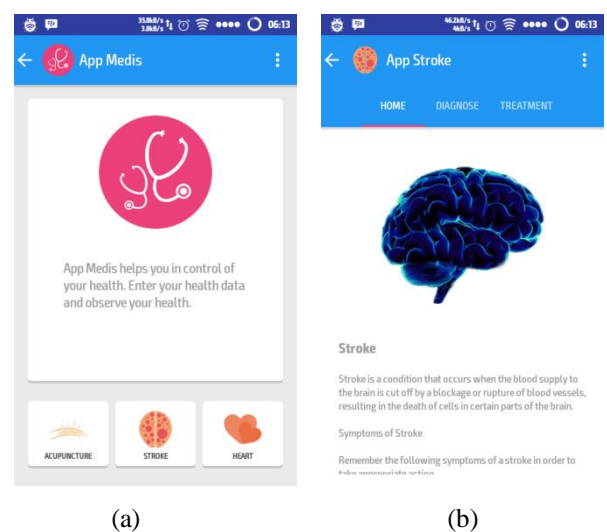
Gambar 4.1 (a) Halaman Awal (b) Main Apps



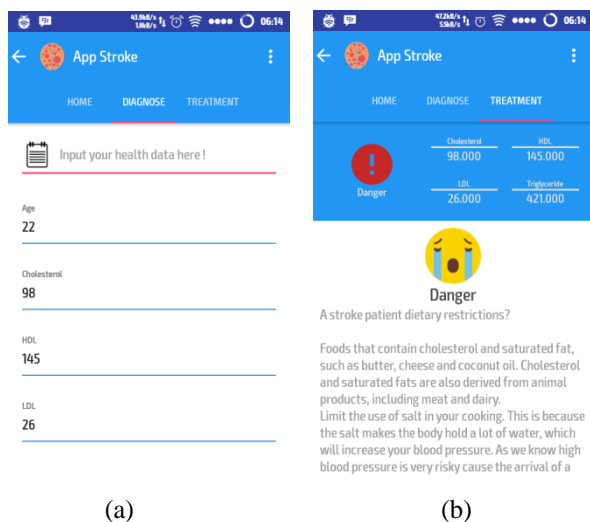
Gambar 4.2 (a) Menu Help (b) Help: Memilih Apps



Gambar 4.3 (a) Help: Login (b) Halaman About



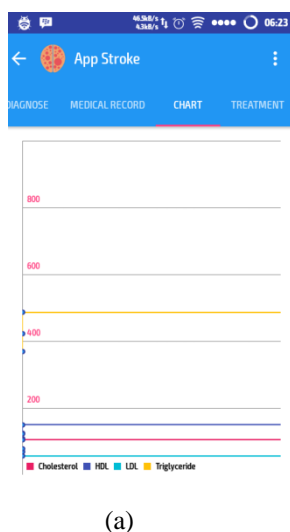
Gambar 4.4 (a) Awal App Medis (b) Home App Stroke



Gambar 4.5 (a) Halaman Diagnosa (b) Halaman Treatment



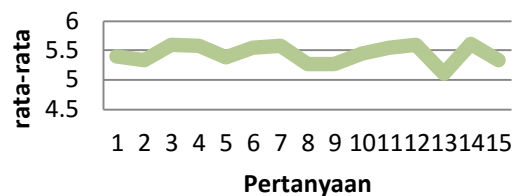
Gambar 4.6 (a) (b) Medical Record



Gambar 4.7 Halaman Chart

Berdasarkan hasil kuisioner yang telah diisi, ditunjukkan hasil rerata dari setiap pertanyaan dalam kuisioner yang telah diberikan kepada pengguna. Hasil rerata tersebut direpresentasikan pada Gambar 4.8.

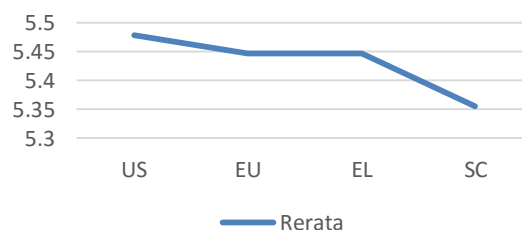
Rerata Pertanyaan



Gambar 4.8 Grafik Rerata Pertanyaan

Dari seluruh pertanyaan yang telah diberikan kepada pengguna, didapatkan nilai rerata terendah dari pertanyaan nomor 13 dengan pertanyaan “Saya puas dengan sistem ini..” Sedangkan nilai rerata tertinggi terdapat pada pertanyaan nomor 14 dengan rincian pertanyaan “Menurut saya aplikasi ini perlu direkomendasikan digunakan oleh tim medis dan masyarakat umum.” Jalannya suatu aplikasi ini dapat diketahui masih terdapat banyak kekurangan dikarenakan kepuasan yang didapatkan oleh seorang pengguna saat menjalankan aplikasi masih belum maksimal. Hal ini dikarenakan, pengguna memerlukan waktu yang lebih untuk memahami mengoperasikan aplikasi. Namun tujuan dan manfaat dari pembuatan aplikasi ini dibutuhkan sehingga perlu untuk digunakan oleh masyarakat umum maupun tim medis terkait. Selain itu dengan penggunaan aplikasi *Lide* pengguna dapat mengetahui lebih cepat tentang tingkat resiko penyakit stroke yang diderita.

Rerata Parameter Modul



Gambar 4.9 Grafik Rerata Modul

Gambar 4.9 diatas merupakan diagram yang didapatkan dari parameter pertanyaan *usefulness* (US), *ease of use* (EU), *ease of learning* (EL), dan *satisfaction* (SC). Jika dilihat dari diagram rerata parameter modul dapat terlihat nilai tertinggi adalah *ease of learning* (EL). Hal ini menunjukkan bahwa aplikasi *Lide* terbuti dapat dipelajari dengan mudah oleh pengguna. Dengan parameter modul terendah terletak pada *satisfaction* (SC) dikarenakan tingkat

kenyamanan maupun kepuasan pengguna yang kurang terhadap aplikasi. Kenyamanan pengguna masih terlihat kurang dikarenakan kurangnya pemahaman dalam pengoperasian aplikasi sehingga terjadi kebingungan saat menjalankan aplikasi stroke.

5. KESIMPULAN & SARAN

Pada penelitian ini telah berhasil mengimplementasikan metode *Support Vector Machine* dan *Particle Swarm Optimization*. Aplikasi *Lide* yang bertujuan untuk mendiagnosa tingkat resiko penyakit stroke dapat dipahami dengan mudah sesuai dengan hasil kuisioner yang didapat dengan rerata 214,181 (*ease of learning*). Namun pengembangan aplikasi sangat diharapkan dengan menggunakan sistem cerdas yang ditanamkan pada server, sehingga proses untuk mendapatkan hasil deteksi akan lebih cepat. Berbagai metode optimasi dalam perhitungan untuk mendapatkan hasil akurasi yang lebih baik. Sehingga penerapan dari metode optimasi seperti GA, Big Colony, dan lainnya dapat diterapkan untuk pengembangan aplikasi. Pengembangan aplikasi pada sistem terdistribusi (*parallel processing*) diperlukan sehingga penggunaan parameter pada setiap program dapat memberikan hasil yang optimal.

6. DAFTAR PUSTAKA

- Chen, H., Yang, B. & Wang, G. 2011. A Novel Bankruptcy Prediction Model Based on an Adaptive Fuzzy K-Nearest Neighbor Method. *College of Computer Science and Technology, Jilin University, Changchun 130012, China*.
- Dinata, C. A., Safrita, Y. & Sastri, S., 2013. Gambaran Faktor Risiko dan Tipe Stroke pada Pasien rawat Inap di Bagian Penyakit Dalam RSUD Solok Selatan. *kesehatan Andalas*, 2(2), pp. 57-61.
- Du, Y., Lu, X. & Hu, C., 2014. Parameters optimization of T-S fuzzy classification system using PSO and SVM. Nanjing, s.n.
- Kesehatan., K. K. (2013). *Riset Kesehatan Dasar*.
- Listiana, L. & Purbosari, T. Y., 2012. Kadar Kolesterol Total pada usia 25-60 tahun. pp. 36-40.
- Lu, N. & Jianzhong, H. Y., 2009. Smulated Annealing Theory based Particle Swarm Optimization fo Support Vector Machine Model in Short-term Load Forecasting. Yichang, s.n.
- Marbun, Y. (2013). *Analisa PSO dan GA untuk optimasi penjadwalan matakuliah*. Jakarta: Universitas Indonesia.
- Ratnasari, D. (2013). *Klasifikasi kondisi pasien stroke menggunakan regresi logistik dan SVM*. Surabaya: Institut Teknologi Sepuluh November.
- Soebroto A. A., Cholissodin I., Wihandika R. C., Frestantiya M. T., Arief Z. E., 2015. Prediksi Tinggi Muka Air (TMA) Untuk Deteksi Dini Bencana Banjir Menggunakan SVR-TVIWPSO. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 2(2), 79-86.
- Sun, C. & Gong, D., 2006. Support Vector Machine with PSO Algorithm for Short-term Load Forecasting. Nanjing, s.n.
- Zhang, H. & Mao, H., 2009. Feature Selection for the Stored-grain Insects Based on PSO and SVM. Zhenjiang, s.n.