

Hybrid Classifier System: Support Vector Machines Dikombinasikan dengan K-Nearest Neighbors untuk Menentukan Kelayakan Nasabah Bank dalam Pengajuan Kredit

Hybrid Classifier System: Support Vector Machines Combined with K-Nearest Neighbors to Determine Bank Customer Feasibility in Applying for Credit

S L Br Ginting^{1*}, A A Permana²

^{1,2}Program Studi Teknik Komputer, Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia
Jl. Dipati Ukur No. 112 - 116, Bandung, Indonesia 40132

*email: selvia.lorena@email.unikom.ac.id

ABSTRACT - This research intends to build an application that can analyze bank data and then determine the feasibility in terms of creditworthiness, to avoid non-performing loans in the future. The method used is a hybrid method that combines two Data Mining classification techniques namely Support Vector Machines (SVM) and K-Nearest Neighbors (KNN). SVM works by finding the optimal hyperplane and support vectors. Furthermore, the KNN will classify bank data based on identifying the support vectors. With 2000 training data and 103 testing data: cost parameter values = 0.1, gamma = 2, 1998 support vectors, then with K value = 16 the system gives 88.35% suitable data (91 data from 103). In conclusion, the application can work in terms of helping the credit analysts to recommend prospective customers who deserve loans.

Keywords - application; data mining; hybrid method; SVM-KNN

ABSTRAK - Riset ini dilakukan dengan maksud membangun aplikasi yang dapat menganalisis data nasabah bank kemudian menentukan kelayakan nasabah tersebut dalam hal pemberian pinjaman, agar terhindar dari masalah kredit macet dikemudian hari. Metode yang digunakan adalah metode hybrid yang menggabungkan 2 teknik klasifikasi Data Mining yaitu Support Vector Machines (SVM) dan K-Nearest Neighbors (KNN). SVM bekerja dengan cara menemukan hyperplane yang optimal dan support vector. Lebih lanjut, algoritma KNN akan melakukan klasifikasi data nasabah bank berdasarkan pengidentifikasian support vector tersebut. Dengan 2000 data latih dan 103 data uji: nilai parameter $cost=0,1$, $gamma=2$, sistem mengidentifikasi 1998 support vector, kemudian dengan nilai $K=16$ sistem memberikan hasil 88,35% data yang cocok (91 data dari 103). Dapat disimpulkan bahwa aplikasi ini bekerja dengan cukup baik dan dapat membantu credit analyst dalam merekomendasikan nasabah yang layak memperoleh pinjaman.

Kata Kunci - aplikasi; data mining; klasifikasi; metode hybrid; SVM-KNN

1. PENDAHULUAN

Dalam dunia perbankan, salah satu layanan yang ditawarkan kepada pelanggan adalah pinjaman atau kredit. Pinjaman atau kredit ini selanjutnya diberikan kepada nasabah yang memang sesuai dengan kriteria yang ditetapkan oleh pihak bank sehingga bisa dipastikan cicilannya tidak akan menunggak di

kemudian hari. Namun, dengan banyaknya jumlah nasabah yang mengajukan kredit, seringkali analisis kredit menjadi ceroboh dalam menentukan nasabah yang layak mendapatkan kredit [1], [2]. Konsekuensinya adalah baik nasabah perorangan maupun perusahaan melakukan penunggakan atau tidak membayar sesuai dengan waktu yang telah ditetapkan. Selanjutnya kondisi ini disebut kredit

macet. Kredit bermasalah atau sering disebut kredit macet adalah keadaan di mana nasabah tidak membayar minimum pembayaran yang telah jatuh tempo dan sudah melewati 3 bulan penunggakan [3]. Berdasarkan kondisi tersebut, dibutuhkan suatu sistem yang mampu menyeleksi dengan cepat dan memprediksi secara akurat dengan memanfaatkan informasi yang ada (data historis) sebagai tolok ukur dalam pemberian kredit kepada nasabah yang benar-benar layak (memenuhi persyaratan yang ditetapkan) [4]-[7].

Penelitian ini bermaksud membuat sebuah aplikasi dengan menerapkan teknologi di bidang *data mining* yang mampu bekerja menemukan pola-pola data yang tidak diketahui sebelumnya. Tahapan kegiatan yang dilakukan dalam *data mining* adalah meliputi pengumpulan data historis lalu kemudian data historis tersebut dimanfaatkan dengan tujuan menemukan pola atau relasi data set yang berskala besar [8], [9]. Metode yang diterapkan dalam membangun aplikasi ini adalah metode *hybrid* yang menggabungkan dua buah teknik klasifikasi *data mining* yaitu teknik *Support Vector Machines* selanjutnya disebut SVM dan *K-Nearest Neighbors* disingkat dengan KNN. Dalam penelitian lain penggabungan metode ini terbukti meningkatkan metode SVM untuk klasifikasi dikarenakan SVM mengambil keuntungan dari metode KNN berdasarkan sampel uji dalam sebuah ruang fitur dan memberikan prediksi dengan akurasi yang lebih tinggi dibandingkan dengan metode SVM saja atau KNN saja [10]-[12]. Metode ini akan membentuk model klasifikasi setelah angka karakteristik instruksional diberikan yang bertujuan memutuskan kelas yang sesuai untuk sampel data yang dianalisis. Dengan menerapkan metode *hybrid* ini diharapkan dapat mengklasifikasi data nasabah yang mengajukan kredit dalam skala besar sebagai alat bagi *credit analyst* dalam menentukan apakah seorang nasabah memenuhi syarat untuk mendapatkan pinjaman.

Metode *hybrid* ini telah diterapkan pada beberapa penelitian lain. R. Li et al menggabungkan SVM dan KNN untuk membangun model peramalan surya dan terbukti penggabungan metode ini memberikan hasil yang lebih baik [10]. J. S. Raikwal dan K. Saxena menerapkan metode *hybrid* ini untuk menganalisis data nominal pasien (data medis) untuk memprediksi penyakit masa depan [11]. Eksperimen penggabungan SVM dan KNN ini juga telah dilakukan oleh C. Domeniconi dan D. Gunopulos dengan menganalisis skema pembelajaran adaptif dengan menggunakan baik simulasi ataupun *real data set*.

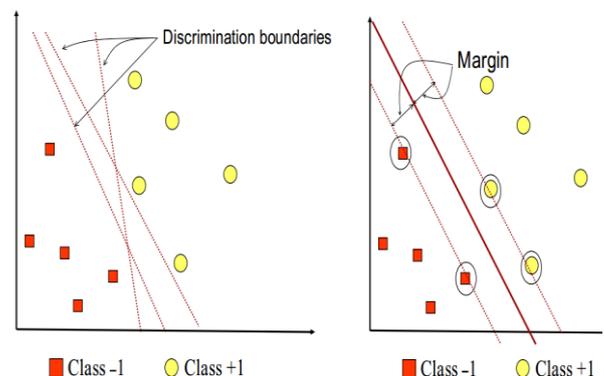
Penelitian ini dilakukan dengan maksud membangun suatu aplikasi yang

mengimplementasikan metode *hybrid* untuk menguji kelayakan nasabah bank baik perorangan maupun perusahaan dalam pemberian kredit. Dengan bantuan aplikasi ini, diharapkan bank dapat lebih cepat memberi kepastian pinjaman dan juga dengan akurat. Aplikasi ini diharapkan dapat menekan biaya operasional dengan kemampuan untuk menentukan kredit area yang kemudian menjadi fokus bank di masa

2. METODE DAN BAHAN

2.1. Support Vector Machines (SVM)

Ide dasar dari metode SVM ini adalah untuk mencari hyperplane atau pemisah yang optimal antara satu kelas dengan kelas lainnya [13]. Metode ini mengklasifikasikan data baik itu data linier atau non-linier. Hyperplane yang dicari adalah yang optimal yang berada diantara satu kelas dengan kelas lain dan caranya adalah menghitung lebar margin secara maksimal. Margin adalah jarak dari hyperplane atau bidang pemisah optimal terhadap poin-poin terdekat yang berada dimasing-masing kelas. Poin-poin yang paling dekat disebut *support vector* [13], [14].



Gambar 1. SVM mencoba mendapatkan hyperplane terbaik pemisah kedua kelas

Gambar 1 (sebelah kiri), kelas positif ditandai dengan +1 (dengan simbol lingkaran) sedangkan kelas negatif ditandai dengan *class -1* (dengan simbol kotak). Proses belajar dalam klasifikasi ini diartikan sebagai satu usaha menemukan garis (*hyperplane*) yang memisahkan antara kedua kelas tersebut. Disamping itu, ada beberapa alternatif garis pemisah (*discrimination boundaries*) ditunjukkan pada gambar 1 yang di sebelah kanan. Pola yang paling dekat dengan garis itu disebut *support vector*. Pada gambar 1 yang disebalah kanan, garis tegas menampilkan *hyperplane* terbaik yang berada persis diantara kedua kelas, disamping itu titik merah dan

kuning yang ditandai oleh lingkaran berwarna hitam merupakan *support vector* [13].

2.2 K-Nearest Neighbors (KNN)

KNN merupakan metode yang menerapkan supervised learning [15]. Metode ini didasarkan pada pembelajaran analogi, yaitu membandingkan data testing dengan data training [16]. KNN melakukan klasifikasi bukan berdasarkan model melainkan berdasarkan memori. Algoritma ini bekerja berdasarkan mencari jarak ketetanggaan terdekat ke sampel baru dengan menggunakan rumus *Euclidean distance*. Formula ini berfungsi untuk mencari *distance* atau jarak antara dua titik di dalam ruang dua dimensi yang dapat dilihat dalam persamaan (1) [17].

$$d(x_1, x_2) = ||x_1 - x_2|| = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2} \quad (1)$$

Yang mana x_1 adalah data latih; x_2 adalah data uji

Tahapan-tahapan dalam algoritma ini adalah: [16]

1. Tentukan jumlah tetangga terdekat yaitu parameter K
2. Hitung *Query instance* atau kuadrat *Euclidean distance* dari tiap-tiap objek sampel data yang tersedia.
3. Urutkan objek-objek ke dalam kelas yang memiliki nilai *Euclidean distance* terkecil.
4. Kumpulkan kategori Y (klasifikasi *nearest neighbor*)
5. Gunakan kategori Y *nearest neighbor* mayoritas sehingga nilai untuk *query instance* dapat diprediksi.

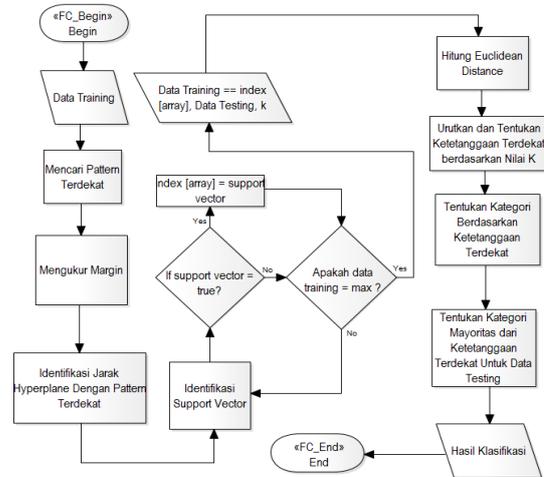
Dalam metode atau algoritma ini, perlu ditentukan nilai K (tetangga terdekat) terlebih dahulu, setelah itu klasifikasi data yang baru dilakukan. Untuk menentukan nilai K perlu pertimbangan yang didasarkan pada besarnya jumlah data yang tersedia serta dimensi yang terbentuk dari data tersebut. Ketika data semakin banyak sebaiknya nilai K semakin kecil. Disamping itu, semakin besar dimensi data, angka K juga semestinya semakin tinggi.

2.3 Sumber Data

Data yang dikaji dalam riset ini adalah data historis bank mulai tahun 2008 sampai 2018 yang diambil dari database sebuah bank negeri di Bandung. Data disimpan di bank dan dapat diperoleh saat pelanggan mengajukan permohonan kredit. Data historis ini dibagi menjadi dua: data latih dan data uji.

2.4 Metode Hybrid

Gambar 2 di bawah ini adalah diagram alir (flowchart) aplikasi secara keseluruhan yang menerapkan metode hybrid.



Gambar 2. Flowchart SVM-KNN

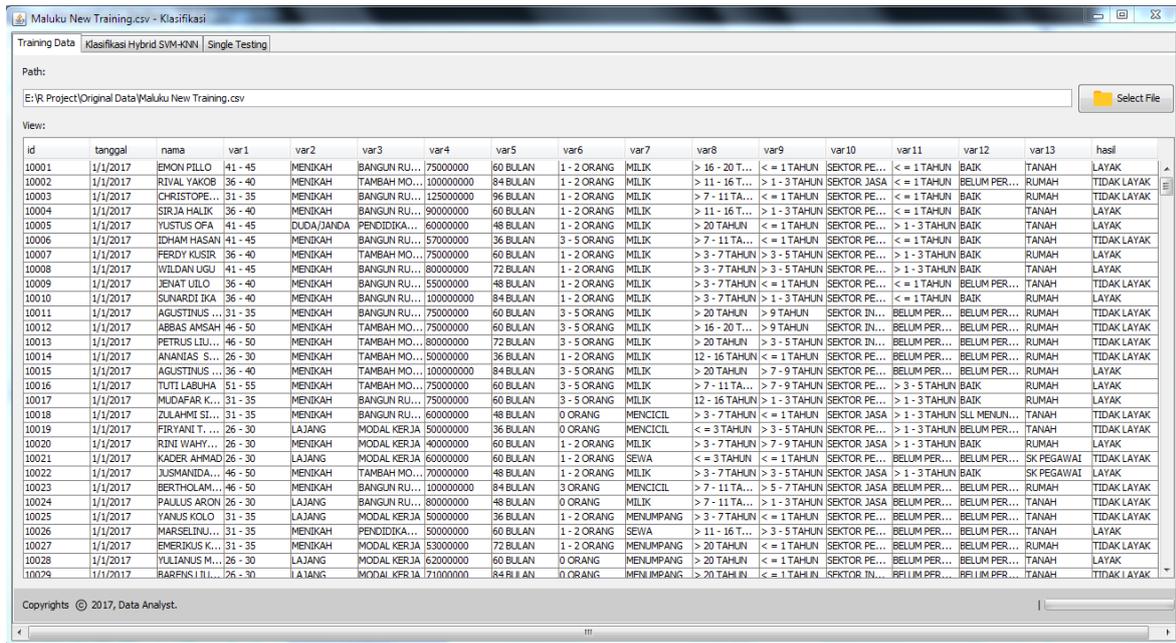
3. HASIL DAN PEMBAHASAN

3.1 Menu Data Training, Menu Hybrid SVM-KNN dan Menu Single Test

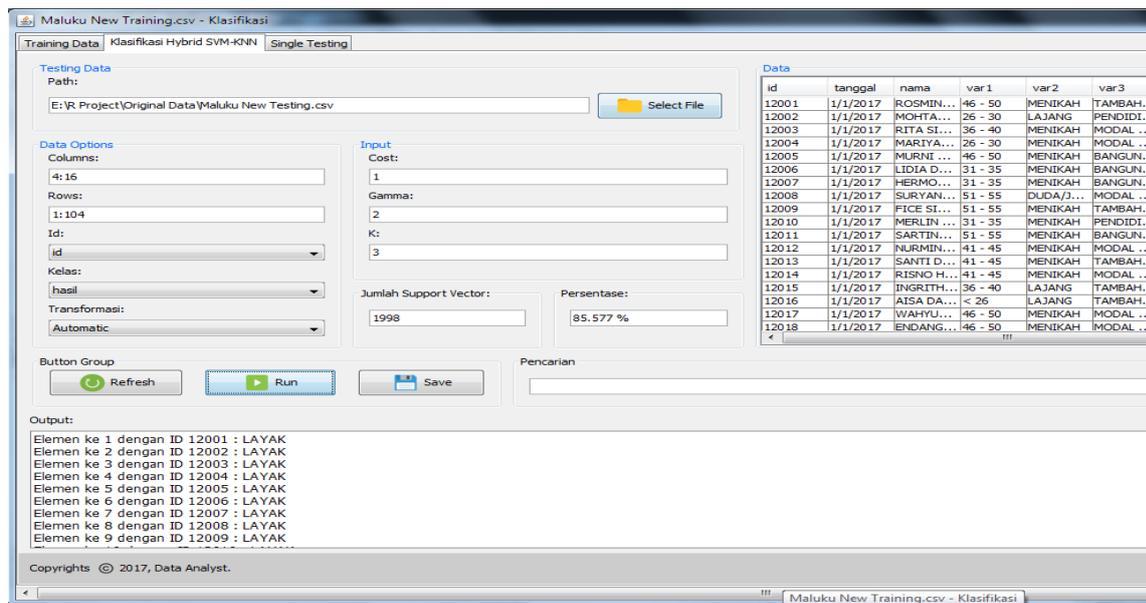
Untuk melakukan klasifikasi data, dibutuhkan data *training* atau data latih. Data latih ini adalah data nasabah yang disimpan dalam suatu tabel dan akan dipanggil pada saat melakukan klasifikasi. Cuplikan data latih akan ditunjukkan pada gambar 3 di bawah ini.

Disamping itu ada menu *Hybrid SVM-KNN* yang akan digunakan *end-user* untuk menentukan layak atau tidaknya nasabah dengan jumlah data test yang banyak (lebih dari satu) dan pengambilan data secara tidak langsung. Ditampilkan pada gambar 4 di bawah ini.

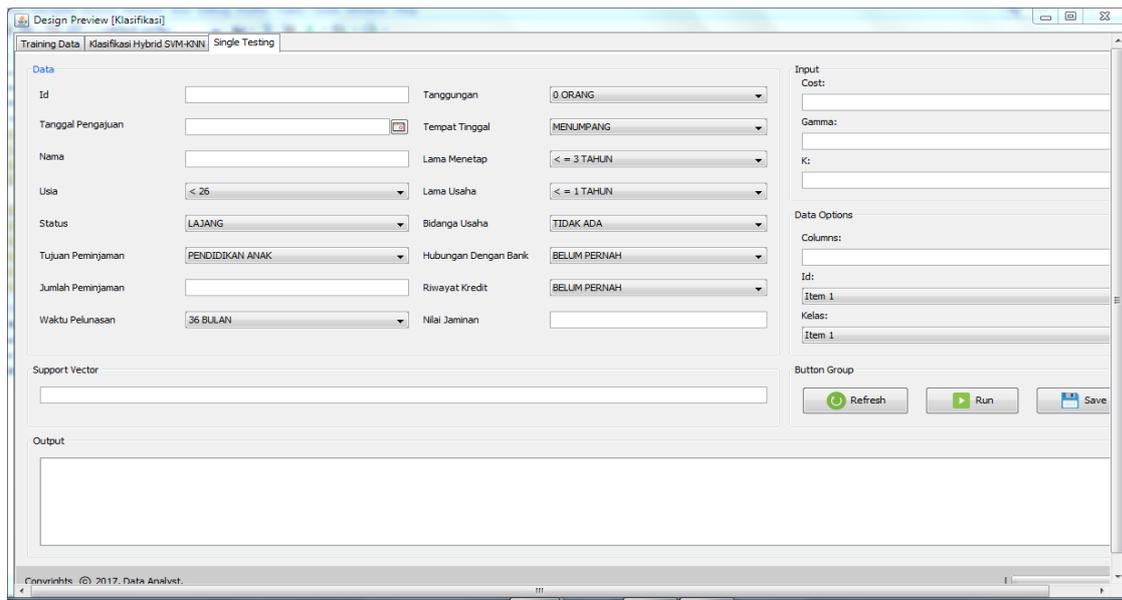
Gambar selanjutnya gambar 5 adalah menu *Single Test* yang memiliki fungsi yang sama dengan menu SVM-KNN. Perbedaannya hanya pada data test yang akan diuji. Pada menu ini data test yang diuji hanya bisa berjumlah satu data dan proses pengambilan data dilakukan secara langsung oleh user dari form/aplikasi nasabah.



Gambar 3. Menu Data Training



Gambar 4. Menu Klasifikasi Hybrid SVM-KNN



Gambar 5. Menu Single Test

3.2. Pengujian Sistem

Pengujian akan dilakukan terhadap data *training* dan data *testing* menggunakan metode *hybrid* (SVM-KNN). Adapun jumlah data latih yang akan diuji adalah 2000 data sedangkan jumlah data uji adalah 103 data. Rencana pengujian ditunjukkan pada tabel 1 berikut ini.

Tabel 1. Rencana Pengujian Sistem Klasifikasi *Hybrid* SVM-KNN

SVM		KNN	Transformasi	
Cost	Gamma	K	Data Uji	
0.1	0.02	2	103	Automatic
10	0.02	4		
0.1	0.2	8		
10	0.2	16		
0.1	2	2		
10	2	4		
0.1	20	8		
10	20	16		
2000		Default	Data Latih	

Akan ditunjukkan beberapa hasil pengujian yang pada makalah ini disajikan secara acak:

1. Pengujian dengan nilai *cost* (*c*) = 10 dan *gamma* (*g*) = 0,2, sistem mengidentifikasi sebanyak 1236 *support vector*. Kemudian sistem melakukan klasifikasi data nasabah menggunakan algoritma KNN. Pada gambar 6 akan terlihat bahwa terdapat persamaan dari hasil klasifikasi

menggunakan aplikasi dengan data asli (tanpa aplikasi). Dengan data latih sebanyak 1236 dan *K* = 16, dengan metode KNN sistem memberikan hasil yang sama sebanyak 90 data dari 103 data. Sedangkan hasil yang tidak sama sebanyak 13 data dari 103 data. Sehingga dapat dinyatakan:

$$hasil = \frac{90}{103} * 100\% = 87,379\%$$

Tingkat keberhasilan sisteem pada pengujian ini adalah sebesar 87,379% dengan data uji sebanyak 103 data.

2. Pengujian dengan nilai *cost* = 0,1 dan *gamma* = 2, dengan menggunakan transformasi data secara otomatis, sistem mengidentifikasi sebanyak 1998 *support vector*. Kemudian sistem melakukan klasifikasi data nasabah menggunakan algoritma KNN. Pada gambar 7 akan terlihat bahwa terdapat persamaan dari hasil klasifikasi menggunakan aplikasi dengan data asli (tanpa aplikasi). Dengan data latih sebanyak 1998 dan *K* = 16, dengan metode KNN sistem memberikan hasil yang sama sebanyak 91 data dari 103 data. Sedangkan hasil yang tidak sama sebanyak 12 data dari 103 data. Sehingga dapat dinyatakan:

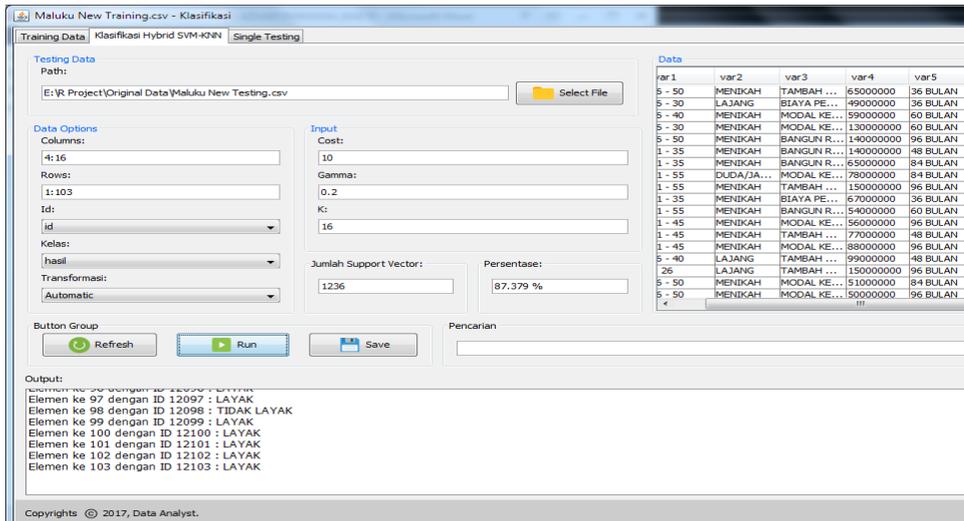
$$hasil = \frac{91}{103} * 100\% = 88,35\%$$

3. Pengujian dengan nilai *cost* = 10 dan *gamma* =

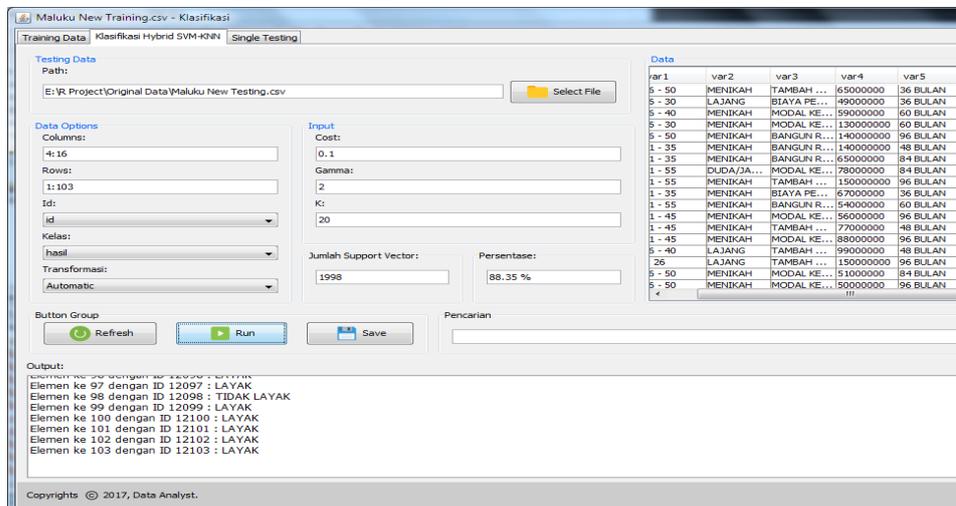
0,2, sistem mengidentifikasi sebanyak 1997 support vector. Kemudian sistem melakukan klasifikasi data nasabah menggunakan algoritma KNN. Pada gambar 8 akan terlihat bahwa terdapat persamaan dari hasil klasifikasi menggunakan aplikasi dengan data asli (tanpa aplikasi). Dengan data latih sebanyak 1997 dan $K = 24$, dengan metode KNN sistem

memberikan hasil yang sama sebanyak 91 data dari 103 data. Sedangkan hasil yang tidak sama sebanyak 12 data dari 103 data. Sehingga dapat dinyatakan:

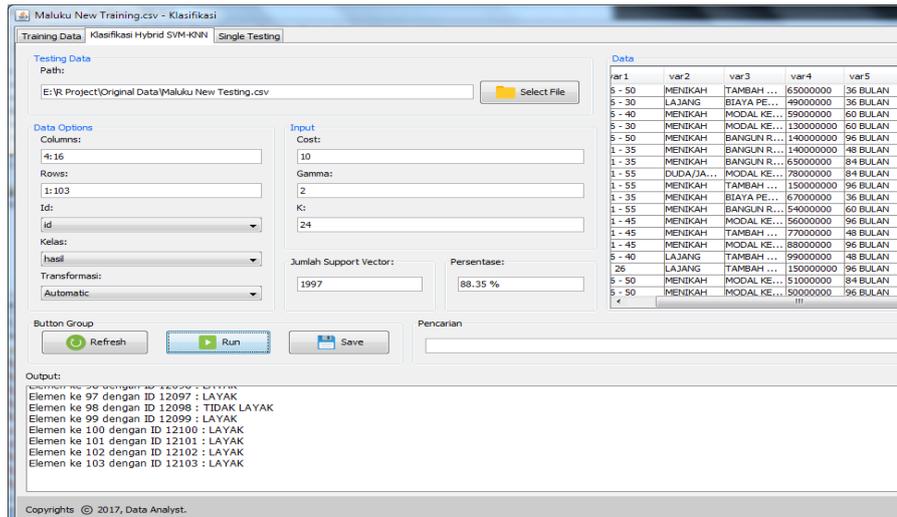
$$\text{hasil} = \frac{91}{103} * 100\% = 88,35\%$$



Gambar 6. Pengujian dengan Nilai $c = 10$, $g=0,2$ dan $K=16$



Gambar 7. Pengujian dengan Nilai $c = 0,1$, $g=2$ dan $K=16$



Gambar 8. Pengujian dengan Nilai $c = 10$, $g=0,2$ dan $K=24$

Tabel 2. Hasil Pengujian Sistem Klasifikasi Data Mining

Parameter			Suppor Vector	Persentase (%)
Cost	Gamma	K		
0.1	0.02	2	895	67,961
10	0.02	4	673	58,252
0.1	0.2	8	1178	87,379
10	0.2	16	1236	87,379
0.1	2	20	1998	88,35
10	2	24	1997	88,35
0.1	20	32	1998	85,437
10	20	64	1997	84,466

Dari tabel 2 di atas, pengujian pertama dan kedua memperoleh persentase keberhasilan adalah sebesar 67,961 %. Kemudian pada dua pengujian selanjutnya diperoleh persentase keberhasilan sebesar 87,379 %. Dapat dilihat bahwa jumlah *support vector* pada kedua pengujian sangat berbeda, oleh sebab itu hasil pengujian diantara keduanya yang terbaik adalah pengujian ketiga dengan *cost* sebesar 0,1, *gamma* sebesar 0,2 dan *K* sebesar 8. Namun dalam dua pengujian selanjutnya diperoleh persentase keberhasilan sebesar 88,35%. Lebih lanjut dua pengujian terakhir menghasilkan nilai persentase keberhasilan yang cenderung menurun. Jadi dapat diambil kesimpulan bahwa untuk menghasilkan

persentase keberhasilan tertinggi adalah dengan *cost* sebesar 0,1 atau 10, *gamma* sebesar 2 dan *K* sebesar 20 atau 24 yang mengindikasikan bahwa nilai dari pada parameter tersebut adalah yang paling optimal dari semuanya.

4. KESIMPULAN

Penerapan metode *hybrid* dilakukan dengan cara menerapkan metode SVM terlebih dahulu dengan menghitung jumlah *support vector*. Kemudian jumlah total dari *support vector* ini akan dijadikan pengganti data *training* yang menjadi masukan bagi metode KNN. Setelah dilakukan beberapa kali pengujian maka diambil sampel dari pengujian tersebut sebanyak 8 kali. Persentase keberhasilan tertinggi dari 8 kali pengujian ini adalah 88,35%. Artinya ada sebesar 88,35% kesamaan data nasabah yang layak diberikan pinjaman yang dihasilkan oleh aplikasi dibandingkan dengan tanpa menggunakan aplikasi. Dari persentase ini mengindikasikan bahwa aplikasi ini dapat diterima dan bisa dipertimbangkan oleh pihak *credit analyst* untuk mereka gunakan sehari-hari dalam menganalisis data nasabah baru yang mengajukan pinjaman. Dengan bantuan sistem atau aplikasi ini pekerjaan para *stakeholder* bank dapat lebih cepat dan efisien. Disisi lain, dari penelitian ini ditemukan faktor-faktor yang mempengaruhi hasil dari klasifikasi yaitu parameter *cost* dan *gamma* yang merupakan parameter pada metode SVM dan parameter *K* pada metode KNN. Ketika nilai *cost*, *gamma* dan *K* sama-sama diperbesar maka persentase kesamaan data latih yang diuji oleh sistem dengan tanpa diuji oleh sistem akan menjadi lebih tinggi. Hal ini juga tentunya dipengaruhi oleh besarnya jumlah data dan seberapa bervariasi data

tersebut.

DAFTAR PUSTAKA

- [1] N. M. D. Widiantari, I. W. Suwendra, and F. Yudiaatmaja, "PENGARUH PENILAIAN KREDIT TERHADAP KEPUTUSAN PEMBERIAN KREDIT PADA BPR," *J. Jur. Manaj.*, vol. 2, no. 1, 2014.
- [2] Kasmir, *Manajemen Perbankan*. Raja Grafindo Persada, 2007.
- [3] M. S. P. Hasibuan, "Dasar-dasar perbankan," *Jakarta PT. Bumi Aksara*, 2001.
- [4] S. L. B. Ginting, J. Adler, Y. R. Ginting, and A. H. Kurniadi, "The Development of Bank Application for Debtors Selection by Using Naïve Bayes Classifier Technique," in *IOP Conference Series: Materials Science and Engineering*, 2018, vol. 407, no. 1, p. 12177.
- [5] V. Bhambri, "Application of data mining in banking sector," *IJCST*, vol. 2, no. 2, pp. 199–202, 2011.
- [6] H. Ince and B. Aktan, "A comparison of data mining techniques for credit scoring in banking: A managerial perspective," *J. Bus. Econ. Manag.*, vol. 10, no. 3, pp. 233–240, 2009.
- [7] K. Chitra and B. Subashini, "Data mining techniques and its applications in banking sector," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 8, pp. 219–226, 2013.
- [8] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
- [9] P.-N. Tan, *Introduction to data mining*. Pearson Education India, 2007.
- [10] R. Li, H.-N. Wang, H. He, Y.-M. Cui, and Z.-L. Du, "Support vector machine combined with K-nearest neighbors for solar flare forecasting," *Chinese J. Astron. Astrophys.*, vol. 7, no. 3, p. 441, 2007.
- [11] J. S. Raikwal and K. Saxena, "Performance evaluation of SVM and k-nearest neighbor algorithm over medical data set," *Int. J. Comput. Appl.*, vol. 50, no. 14, 2012.
- [12] J. KIM¹, B.-S. Kim, and S. Savarese, "Comparing image classification methods: K-nearest-neighbor and support-vector-machines," *Ann Arbor*, vol. 1001, pp. 42122–48109, 2012.
- [13] L. Wang, *Support vector machines: theory and applications*, vol. 177. Springer Science & Business Media, 2005.
- [14] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
- [15] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Advances in neural information processing systems*, 2006, pp. 1473–1480.
- [16] Y. Wu, K. Ianakiev, and V. Govindaraju, "Improved k-nearest neighbor classification," *Pattern Recognit.*, vol. 35, no. 10, pp. 2311–2318, 2002.
- [17] M.-L. Zhang and Z.-H. Zhou, "A k-nearest neighbor based algorithm for multi-label classification," in *Granular Computing, 2005 IEEE International Conference on*, 2005, vol. 2, pp. 718–721.