

Seleksi Fitur Dengan *Particle Swarm Optimization* Untuk Pengenalan Pola Wajah Menggunakan *Naive Bayes* (Studi Kasus Pada Mahasiswa Universitas Brawijaya Fakultas Ilmu Komputer Gedung A)

Satria Habiburrahman Fathul Hakim¹, Imam Cholissodin², Agus Wahyu Widodo³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya

Email: ¹satriatife@gmail.com, ²imamcs@ub.ac.id, ³a_wahyu_w@ub.ac.id

Abstrak

Sistem presensi mahasiswa pada Fakultas Ilmu Komputer, Universitas Brawijaya masih menggunakan sistem manual yang sangat rawan disalahgunakan oleh mahasiswa itu sendiri seperti menitipkan absen pada temannya. Oleh karena itu dibutuhkan suatu sistem yang telah terdigitalisasi dan juga cepat dalam mencari solusi permasalahan. Metode optimasi merupakan metode untuk pencarian solusi yang lebih cepat. Pada penelitian kali ini peneliti menggunakan metode *Particle Swarm Optimization* (PSO), yang mana metode tersebut terinspirasi dari perilaku sosial gerakan kawanan burung pada kehidupan kesehariannya. Sedangkan metode klasifikasi adalah metode yang sangat erat kaitannya dengan hipotesis probabilitas. Jadi 2 metode ini akan memiliki fungsi yang berbeda dalam pengenalan wajah pada presensi mahasiswa tersebut dimana PSO disini adalah sebagai seleksi fitur dan *Naive Bayes* disini sebagai mesin klasifikasi sekaligus juga mempunyai fungsi untuk mendapatkan *fitness*nya. Pada pengujian pengaruh iterasi didapatkan kesimpulan bahwa iterasi dengan total nilai *fitness* terbaik pada jumlah partikel ke 38 yaitu 13,38, kemudian pada pengujian pengaruh jumlah iterasi didapatkan kesimpulan bahwa total *fitness* terbesar berada pada iterasi 190 yaitu 36,799, dengan kata lain semakin besar iterasi maka *fitness* semakin baik pula dan pengujian yang terakhir yaitu pada bobot inersia berada, *fitness* tertinggi pada bobot inersia 1,2 yaitu 1,588.

Kata Kunci: *Particle Swarm Optimization*, *Naive Bayes*, metode optimasi, wajah

Abstract

The Presence system of students in the Faculty of Computer Science, Brawijaya University is still using the manual system that is very prone to be misused by the students as entrusted that presence to his friend. Therefore we need a system that has been digitized and also fast in finding solution problem. Optimization method is a method of searching for faster solutions. For this time the researchers is using the Particle Swarm Optimization (PSO) method, that method was inspired by the social behavior of bird movements in their daily lives. While the method of classification is a method that is closely related to the probability hypothesis. So there are 2 methods and have different functions in facial recognition at the student presences where PSO here is as a feature selection and Naive Bayes here as a classification engine as well as a function to get fitness. In the test results obtained that iteration with the best total fitness value is on the number of particles 38 with the highest total fitness is 13,38, then on testing the effect of the number of iterations obtained the conclusion that the largest total fitness is at iteration 190 is 36,799, in other words the greater of iteration the fitness is also better and the last test is on testing for the weight of inertia is 1,2 with the highest total fitness result is 1,588.

Keywords: *Particle Swarm Optimization*, *Naive Bayes*, optimization method, faces

1. PENDAHULUAN

Dari hasil pengamatan penulis pada sistem presensi mahasiswa yang dilakukan oleh bagian akademik Fakultas Ilmu Komputer Universitas Brawijaya masih menggunakan sistem manual

berupa tanda tangan pada buku presensi, hal ini sangat rentan terhadap kecurangan pada presensi harian. Hal ini terbukti pula pada saat penulis melakukan pengamatan langsung saat berada di dalam sebuah kelas yang berlokasi di gedung A FILKOM pada saat perkuliahan manajemen

industri dan teknologi informasi tanggal 4 Mei 2016. Pada saat itu perkuliahan dimulai pada pukul 13.00 WIB sampai dengan selesainya mata kuliah pukul 15.30 WIB ada 35 mahasiswa yang hadir mengikuti mata kuliah, namun pada buku presensi ada 40 anak yang mengisi presensi kehadiran. Jadi dapat disimpulkan bahwa ada beberapa orang mahasiswa yang tidak hadir tersebut menitipkan presensi pada temannya. Hal ini terjadi dikarenakan faktor sistem yang masih belum terdigitalisasi dan terautomatisasi mendorong peneliti untuk menciptakan suatu sistem yang mana nantinya wajah sebagai objek untuk dapat mengidentifikasi presensi pada mahasiswa.

Wajah atau *face* adalah bagian tubuh yang berada pada kepala bagian depan manusia yang meliputi wilayah dahi, hidung, alis, pipi dan dagu. Setiap orang sangat dipastikan memiliki karakteristik wajah yang berbeda mulai dari ekspresi, bentuk ataupun karakteristik wajah, hal itulah yang menjadi landasan bahwasannya wajah termasuk objek dari bidang biometrik. Biometrik merupakan sebuah metode yang terautomatisasi yang dapat mengenal seseorang berdasarkan karakteristik fisik tertentu seperti iris mata, sidik jari, tanda tangan dan suara (Foka, 2002).

Wajah dapat diproses dengan pola pengenalan wajah. Pendeteksian wajah merupakan langkah paling awal yang harus dilakukan dalam melakukan pengidentifikasian suatu wajah. Pendeteksian suatu wajah yang ideal harus dapat mengidentifikasi maupun menemukan lokasi dari luas semua wajah yang ada pada sebuah gambar dengan tidak memperhatikan skala, pose, umur, ekspresi, maupun orientasi (Jain, 2000). Pendeteksian wajah dapat dilakukan dengan cara segmentasi pada area citra wajah dengan bagian latar (*background*). Dalam aplikasi berbasis *desktop* komputer atau *device* lainnya pengenalan wajah dapat diproses dengan alur sesuai algoritme dan metode tertentu, agar diperoleh kualitas pengenalan wajah yang baik dibutuhkan suatu kondisi lingkungan yang mendukung untuk dilakukannya uji coba atau dipakainya sistem tersebut diantaranya dengan indeks cahaya, warna kulit, sudut-sudut pada saat pengambilan gambar wajah dan ukuran *pixel* kamera saat pengambilan gambar wajah berlangsung (Putra, 2013).

Namun sistem itu belum banyak digunakan di kalangan umum seperti sekolah, universitas, perkantoran dan perusahaan. Disamping

permasalahan tersebut ada permasalahan yang sangat penting yaitu permasalahan sistem yang harus mendapatkan solusi permasalahan dengan lebih cepat. Oleh karena itu dibutuhkan suatu sistem yang dapat menyelesaikan masalah tersebut dengan cara mengimplementasikan metode optimasi. Ada banyak teknik untuk optimasi namun pada penelitian kali ini penulis menggunakan teknik optimasi dengan algoritme *Particle Swarm Optimization*, dikarenakan algoritme PSO memiliki proses yang lebih sederhana dibandingkan dengan algoritme optimasi lainnya seperti algoritme genetika. Hal ini dikarenakan algoritme PSO tidak memiliki proses *crossover* dan mutasi, selain itu parameter algoritme PSO juga lebih sedikit daripada algoritme genetika (Khusna, 2016).

PSO pada penelitian kali ini berfungsi sebagai pemberi referensi seleksi fitur yang terbaik dan layak untuk diproses agar mendapatkan hasil yang maksimal. PSO akan mencari berdasarkan ukuran populasi (*swarm*) dari tiap individu (partikel) dengan syarat harus diperbaharui pada tiap iterasinya. PSO akan punya beberapa parameter berupa posisi, kecepatan, kecepatan *max*, konstanta kecepatan dan bobot inersia. PSO punya kelebihan untuk optimasi yang lebih cepat dan juga memiliki tingkat konvergensi yang stabil (Park, 2009). PSO juga mempunyai kelebihan yang lain yaitu mempunyai konsep sederhana, mudah diimplementasikan, dan efisien dalam perhitungan jika dibandingkan dengan algoritme matematika dan teknik optimisasi heuristik lainnya (Tuegeh dkk, 2013).

Penelitian selanjutnya pada *Naïve Bayes* penelitian yang dilakukan oleh (Fais dkk, 2014) yang mana dalam menentukan keberhasilan akurasi pada metode *Naïve Bayes* dipengaruhi oleh kualitas dari *preprocessing* data sebelum dilakukannya klasifikasi pada *Naïve Bayes*. Karena pada tiap pengujian pada penelitian tersebut mengalami perbedaan pada tiap adanya perubahan *data training* pada proses hasil klasifikasi.

Sedangkan untuk seleksi fitur warna merujuk kepada penelitian yang dilakukan oleh (Choi, 2011) terdapat 9 fitur warna pada pengenalan wajah yaitu YCbCr, YIQ, HSV, Cie $l^*u^*v^*$, RGB, YUV, XYZ, CIE $L^*a^*b^*$, Cie L^*ch . Pada penelitian tersebut fitur diseleksi berdasarkan kedekatan warna satu dengan yang lain dari data latih sebagai contoh RGB dan HSV akan menghasilkan informasi dari intensitas warna hijau. Kedekatan warna ini dihitung

menggunakan metode KNN (*nearest neighbor*). Pada penelitian yang lain yaitu pendeteksian kulit dimana fitur warna yang terbaik pada penelitian yang dilakukan oleh (Sutanto, 2010) adalah RGB dan YcbCr. Pada penelitian tersebut RGB dan YcbCr memberikan pendeteksian yang lebih tinggi dan juga menghasilkan akurasi sebesar 83%.

Berdasarkan keberhasilan, hasil implementasi dan juga studi pustaka dari penelitian sebelumnya yang menggunakan metode PSO dan juga *Naïve Bayes* penulis memiliki hipotesis bahwa penelitian ini menghasilkan solusi yang optimal terkait dengan hasil *fitness* yang didapatkan pada pengujian nantinya. Penelitian pada kali ini ialah klasifikasi objek citra wajah dengan gabungan dari 2 metode dimana metode PSO disini adalah sebagai penyeleksi fitur yang paling optimal diakarenakan metode PSO ini mampu menyelesaikan masalah yang kompleks dan juga metode yang lebih sederhana dibandingkan dengan metode optimasi lainnya seperti algoritme genetika, sedangkan metode *Naïve Bayes* disini sebagai mesin klasifikasi yang menghasilkan akurasi sebagai *fitness* sebagai hasil keluarannya.

2. LANDASAN KEPUSTAKAAN

2.1 Konsep Pengenalan Pola

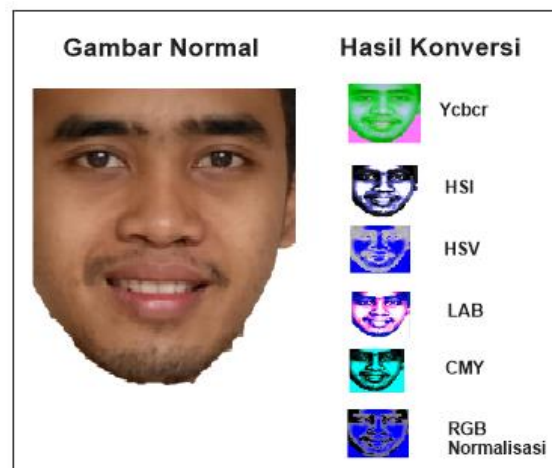
Pola adalah suatu entitas yang sudah terlebih dahulu didefinisikan berdasarkan ciri-cirinya. Dari ciri-ciri itulah yang nantinya akan sangat berguna sebagai pembeda antara pola satu dengan pola yang lain. Suatu pola yang memiliki ciri-ciri yang baik akan memiliki pola dengan pembeda yang sangat tinggi, sehingga pada pengenalan pola berdasarkan pengelompokan ciri tersebut didapatkan nilai keakurasian yang tinggi. Pola berasal dari gabungan berbagai fitur dan merupakan sifat dari salah satu objek (Al Fatta, 2009).

2.2 Pengenalan Wajah (*Face Recognition*)

Wajah merupakan bagian dari kepala bagian depan manusia yang meliputi wilayah dari dahi saapai bagian dagu, termasuk juga bagian rambut, dahi, alis, mata, hidung, pipi, mulut, bibir, gigi, kulit, dan dagu (KBBI, 2009).

Elemen-elemen tersebut yang menjadi pembeda antara wajah seseorang dengan yang lainnya. Selain elemen fisik tersebut ada beberapa faktor lain yang mempengaruhi wajah yaitu: jaringan syaraf dan pembuluh darah,

trauma secara fisik dan hasil pembedahan, gender, ras, pertumbuhan dan usia. Oleh karena itu tidak ada satu wajahpun yang serupa antara individu satu dengan yang lain, bahkan pada manusia kembar identik sekalipun.



Gambar 1. Ilustrasi Gambar Dataset Wajah

2.3 Fitur Seleksi

2.3.1. YcbCr

Warna YcbCr kita bisa juga menyebutnya dengan istilah CCIR 601 (*International Radio Consultative Commite*). YcbCr ini mempunyai range warna dari 0-255. Warna ini awalnya dikembangkan untuk dunia informasi digital berbasis video, sehingga dalam perkembangan video digital tidak lepas dari pengembangan dari model warna tersebut. Pada umumnya model warna YcbCr ini adalah bagian dari ruang warna yang berasal dari transmisi video dan televisi. Warna YcbCr ini juga mempunyai hubungan tetangga yang mirip dengan warna lain yaitu YUV dan YIQ, perbedaan yang terjadi adalah apabila YcbCr adalah sistem warna secara digital sedangkan yang lain adalah sistem warna secara analog. Warna YcbCr ini memisahkan nilai dari RGB agar menjadi informasi *luminance* dan juga *chrominance* yang sangat berguna pada aplikasi kompresi citra gambar. Konversi dari RGB ke YcbCr ini memiliki persamaan sebagai berikut (Ford dan Roberts, 1998):

$$Y = 0.299900.R + 0.58700.G + 0.11400.B$$

$$Cb = -0.16874.R - 0.33126.G + 0.50000.B$$

$$Cr = 0.50000.R - 0.41869.G - 0.08131.B \quad (1)$$

2.3.2 CMY

Warna CMY (*Cyan, Magenta dan Yellow*)

adalah model *subtractive* yang mana hubungannya terjadi dengan adanya penyerapan warna. CMY ini mempunyai range warna dari 0-255. Contoh dari implementasi warna CMY adalah pigmen warna pada cat. Permukaan yang telah dicat oleh warna cyan akan diiluminasi oleh sinar putih, tidak ada sinar merah yang dipantulkan dan juga adanya persamaan untuk warna magenta dengan warna hijau dan juga kuning dengan biru. Konversi dari RGB ke CMY memiliki persamaan sebagai berikut (Ford dan Roberts, 1998):

$$\begin{aligned}C &= 1.0 - R \\M &= 1.0 - G \\Y &= 1.0 - B\end{aligned}\quad (2)$$

2.3.3 HSI

HSI (*hue, saturation, intensity*) adalah bentuk bidang datar dari model solid yang dilakukan pemotongan secara horizontal pada intensitas tertentu. HSI ini mempunyai range warna dari 0-255. *Hue* berasal dari warna merah, *saturation* berasal dari jarak sumbu. Warna permukaan yang solid berasal dari saturasi yang penuh yang mana terdiri atas warna murni dan spektrum tingkat keabuan. Konversi dari RGB ke HSI memiliki persamaan sebagai berikut (Ford dan Roberts, 1998):

$$\begin{aligned}H' &= \cos^{-1} \left(\frac{\frac{1}{2} [R-G] + (R-B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right) \\S &= 1 - 3 / (R+G+B) \min(R,G,B) \\I &= (R+G+B) / 3\end{aligned}\quad (3)$$

2.3.4 HSV

HSV (*Hue, Saturation dan Value*) dimana *hue* adalah warna yang telah direfleksikan maupun dilakukan transmisi dari sebuah objek. HSV ini mempunyai range warna dari 0-255. Nilai *hue* dapat diukur dari lokasi roda standar pada warna yang mana direpresentasikan dalam nilai derajat dengan besaran sudut antara 0° dan 360°. *Hue* dalam pengimplementasiannya dapat melakukan identifikasi nama dari warna merah, orange (jingga) atau hijau. *Saturation* (*chroma*) adalah merupakan ukuran kemurnian dari warna yang terjadi. *Saturation* akan membentuk representasi keabuan pada *hue* dan ukurannya akan berbentuk dalam persentase 0% (*gray*) sampai 100% (*full saturated*). Untuk roda standar warna, *saturation* dari pusat roda yang berbentuk lingkaran ini akan semakin bertambah

dengan menuju ketepian. Untuk *value* mempunyai tingkat kecerahan dari warna yang bervariasi dari *saturation*. Nilai warna ini antara 0-100%. Apabila warna tersebut memiliki nilai 0, maka akan menjadi hitam dan apabila nilai tersebut dinaikkan akan membentuk variansi baru dari warna tersebut. Model warna ini menganut sistem warna yang berasal dari teori warna Ostwald (Ostwald, 1931).

$$S = \frac{(max-min)}{max}$$

$$V = max$$

H berasal dari:

$$\text{If } R = max \text{ and } G = min \text{ (} H = 5 + B' \text{)}$$

$$\text{Else if } R = max \text{ and } G \neq min \text{ (} H = 1 - G' \text{)}$$

$$\text{Else if } G = max \text{ and } B = min \text{ (} H = R' + 1 \text{)}$$

$$\text{Else if } G = max \text{ and } B \neq min \text{ (} H = 3 - B' \text{)}$$

$$\text{Else if } R = max \text{ (} H = 3 + G' \text{)}$$

$$\text{otherwise (} H = 5 - R' \text{)} \quad (4)$$

2.3.5 RGB

RGB adalah model dimana warna berasal dari konsep penambahan kuat dari cahaya primer yaitu *red, green* dan *blue*. Pada satu ruang yang sama apabila tidak adanya cahaya maka suatu ruangan tersebut akan mengalami gelap total. Lalu apabila kita tambahkan cahaya merah di ruangan tersebut maka ruangan itu akan berubah menjadi merah pada RGB menjadi (255,0,0) maka ruangan tersebut hanya dapat terlihat dengan warna merah saja. Dan hal tersebut juga akan sama apabila cahaya yang diberikan akan diganti dengan warna biru ataupun hijau (Permana, 2010). RGB erat kaitannya dengan normalisasi RGB dimana 3 komponen utama yaitu *r, g, dan b* mengandung unsur persentase dari sebuah pixel yang ada digambar. Persamaan untuk normalisasi adalah sebagai berikut (Ford dan Roberts, 1998):

$$r = \frac{R}{R + G + B}$$

$$g = \frac{G}{R + G + B}$$

$$b = \frac{B}{R + G + B} \quad (5)$$

2.3.6 CIE Lab

CIE adalah kepanjangan dari *Commissione Internationale de l'Eclairage* merupakan Komisi Iluminasi Internasional yang mana bertanggung jawab untuk memberikan rekomendasi

internasional pada kasus fotometri dan colorimetri. Pada tahun 1931 lembaga ini berhasil membuat suatu standar warna dari 3 dasar yaitu spesifikasi cahaya sumber (*iluminants*), pengamat dan metodologi untuk digunakan memperoleh nilai pendeskripsian warna. Pada warna CIE ini memanfaatkan 3 koordinat dalam penemuan warna dalam suatu ruangan. Ruang warna itu yaitu:

a. CIE XYZ

Pada saat studi mengenai persepsi warna pertama kali yang berhasil didefinisikan adalah mengenai CIE XYZ (*Introduction to Colour Science, William Andrew Steer & efg's Color Chromaticity Diagrams Lab Report and Delphi source*), yang dibuat pada tahun 1931. Warna CIE XYZ didapat dari rangkaian uji coba yang dilakukan pada 1920 oleh W.David Wright (CIE Color Space) dan John Guild, pada saat itu hasil eksperimen tersebut digabungkan menjadi spesifikasi warna CIE RGB dari turunan warna CIE XYZ.

Perhitungan CIE XYZ ini dimana X bertindak sebagai iluminasi, Z berarti quasi-equal stimulus biru (respon dari kerucut S), dan X adalah kombinasi linear yang berasal dari respon kerucut yang sifatnya ortogonal dari iluminasi dan juga bernilai negatif dengan kata lain bahwa XYZ ini adalah turunan dari respon kerucut L (panjang), M (menengah) dan S (pendek). Dalam penyajian warna XYZ mempunyai standar tersendiri apabila ingin mengkombinasikan penambahan nilai XYZ, lalu ketiga warna tersebut akan dikonversikan ke bentuk RGB dengan matriks berikut.

$$\begin{aligned} X &= 0.412 * R + 0.358 * G + 0.180 * B \\ Y &= 0.213 * R + 0.715 * G + 0.072 * B \\ Z &= 0.019 * R + 0.119 * G + 0.950 * B \end{aligned} \quad (6)$$

Namun CIE XYZ ini mempunyai beberapa kelemahan yaitu sangat sulit pada pengaturan *brightness* sesuai dengan yang diinginkan (Loy, 2002).

b. CIE L*a*b*

LAB sebagai singkatan informal dengan sebutan CIE 1976 (L^* , a^* , b^*). Perbedaan antara Hunter dan warna CIE adalah pada koordinat CIE LAB yang L (*Luminance* atau tingkat kecerahan) a (rentang warna dari hijau sampai merah), dan b (rentang warna dari biru sampai kuning). Pertanda bintang (*) setelah L , a dan b merupakan bagian nama lengkap, karena merupakan perwakilan dari L^* , a^* dan b^* , hal ini ditujukan sebagai pembeda antara CIELAB

dari L Hunter pada CIE XYZ pada Persamaan 6.

Model $L^* a^* b^*$ merupakan model yang berbentuk 3 dimensi, sehingga hanya bisa diwakili dalam ruang 3 dimensi (*3D representations of the $L^*a^*b^*$ gamut, Bruce Lindbloom*). Sedangkan dalam 2 dimensi penggambaran berupa bentuk diagram kromatisitas yang mana merupakan bagian warna dengan *lightness* yang tetap. 1 dimensi merupakan sebagai penentu *lightness*. Namun model ini memiliki kekurangan yaitu representasi visual dari semua warna dalam model yang ada masih belum akurat.

$$\begin{aligned} L^* &= 116 (Y/Y_n)^{1/3} - 16 \\ a^* &= 500 [(X/X_n)^{1/3} - (Y/Y_n)^{1/3}] \\ b^* &= 200 [(Y/Y_n)^{1/3} - (Z/Z_n)^{1/3}] \end{aligned} \quad (7)$$

2.4 Particle Swarm Optimization

Particle swarm optimization adalah suatu algoritma yang banyak terinspirasi dari perilaku sosial hewan seperti burung, lebah dan ikan. Seekor hewan dalam algoritma PSO akan dianggap sebagai partikel. Partikel ini akan dipengaruhi oleh kecerdasan dari individu hewan itu sendiri dan kecerdasan dari partikel lain dalam satu kelompok. Apabila satu partikel menemukan jalan yang tepat dan terpendek menuju ke suatu sumber makanan, maka yang terjadi adalah partikel-partikel lain tersebut akan mengikuti partikel yang telah menemukan jalan yang tepat dan terpendek tadi. Pada algoritma PSO ini memiliki proses perhitungan sebagai berikut:

2.4.1 Inisialisasi Kecepatan Awal

Pada saat inisialisasi kecepatan iterasi awal ini (iterasi ke-0) semua kecepatan partikel haruslah bernilai = 0 atau kita bisa lihat Persamaan 8.

$$(V_{ij}(t)=0) \quad (8)$$

2.4.2 Inisialisasi Posisi Awal Partikel

Posisi awal dari sebuah partikel seharusnya berada pada range $[x_{\min}, x_{\max}]$. Jika posisi yang dihasilkan kurang dari x_{\min} maka posisi harus dikembalikan lagi ke x_{\min} , dan jika posisi melebihi dari x_{\max} maka posisi harus dikembalikan lagi ke x_{\max} . Formula untuk melakukan inisialisasi sebuah partikel dengan range dimensi dapat dilihat pada Persamaan 9.

$$x_{ij}(t) = x_{\min j} + r(x_{ij} - x_{ij}) \quad (9)$$

2.4.3 Inisialisasi *Pbest* dan *Gbest* Awal

Pada inisialisasi *Pbest* iterasi ke-0, nilai partikel ini dapat disamakan nilainya dengan nilai Inisialisasi awal partikel.

$$(Pbest_{i,j}(t) = x_{i,j}(t)) \quad (10)$$

2.4.4 Update Kecepatan

Update kecepatan berguna dalam penentuan ke arah mana perpindahan posisi partikel yang berada pada populasi tertentu. Dimana nilai bobot inersia (w) telah diketahui, untuk koefisien akselerasi 1 dan 2 (c_1 dan c_2) telah diketahui, sedangkan untuk nilai *random* pada r_1 dan r_2 adalah bilangan yang dipilih secara acak dalam interval 0 sampai dengan 1.

$$v_{i,j}^{t+1} = w.v_{i,j}^t + c_1.r_1(Pbest_{i,j}^t - x_{i,j}^t) + c_2.r_2(Gbest_{g,j}^t - x_{i,j}^t) \quad (11)$$

2.4.5 Update Posisi

Apabila kecepatan baru telah didapatkan maka dilakukan proses *update* posisi dengan cara menghitung nilai *sigmoid* berdasarkan Persamaan 12 berikut.

$$sig(v_{i,j}^t) = \frac{1}{1 + e^{-v_{i,j}^t}} \quad (12)$$

Setelah nilai *sigmoid* pada seluruh partikel ke- i dan dimensi ke- j didapatkan maka dilakukan perbandingan antara nilai pada *update* kecepatan terhadap nilai acak yang telah dibuat sebanyak partikel ke- i dimensi ke- j . Apabila nilai *random* lebih kecil dari nilai *sigmoid* maka posisi terbaru bernilai = 1 dan sebaliknya apabila nilai *random* lebih besar dari nilai *sigmoid* maka nilai posisi terbaru bernilai = 0. Hal tersebut dapat dilihat pada Persamaan 13.

$$x_{i,j}^{t+1} = \begin{cases} \text{if } rand[0,1] < sig(v_{i,j}^{t+1}) = 1 \\ \text{else} \\ = 0 \end{cases} \quad (13)$$

2.4.6 Kondisi Berhenti

Kondisi berhenti dalam PSO adalah syarat yang digunakan dalam mengakhiri iterasi pencarian. Syarat-syarat tersebut akan dijelaskan sebagai berikut:

1. Suatu iterasi akan terhenti jika telah mencapai nilai maksimum

2. Suatu iterasi akan terhenti jika telah ditemukan solusi yang telah memenuhi kriteria yang ada.
3. Suatu iterasi akan terhenti apabila tidak adanya perubahan nilai (konvergen).
4. Suatu iterasi akan terhenti apabila nilai radius *swarm* yang telah dinormalisasi mendekati 0.
5. Suatu iterasi akan terhenti apabila grafik fungsi obyektif telah mendekati nilai 0 seiring dengan pertambahan iterasi.

2.5 Naïve Bayes

Naïve Bayes Classifier (NBC) adalah salah satu aspek dari pengenalan pola (*pattern recognition*) yang sudah merujuk pada statistik yang fundamental. *Naïve Bayes* adalah teknik yang prediksi berdasarkan probabilitas. Implementasi dari teorema bayes ini adalah berdasarkan asumsi independensi yang dominan (Eko Prasetyo, 2012).

Dalam teori bayes ada beberapa prinsip ada beberapa prinsip ketidaktergantungan yang kuat untuk tiap fitur pada sebuah data yang tidak berhubungan dengan adanya fitur lain pada data yang sama. Pada hipotesis *Naïve Bayes* adalah merupakan label identitas kelas yang telah menjadi target pada pemetaan klasifikasi dengan telah terkait pada korelasi hipotesis, dan bukti tersebut dapat berupa fitur-fitur yang telah menjadi input pada model klasifikasi. Persamaan pada klasifikasi *Naïve Bayes* adalah sebagai berikut:

$$P(Y|X) = \frac{P(Y) \prod_{i=1}^d P(X_i|Y)}{P(X)} \quad (14)$$

3. PERANCANGAN

3.1 Formulasi Permasalahan

Seorang mahasiswa FILKOM UB dimisalkan bernama A akan diidentifikasi wajahnya. Pertama masukkan gambar wajah pada sistem untuk dilakukannya konversi nilai gambar kedalam beberapa fitur yaitu: HSV, YCbCr, RGB, HSI dan CIELab. Setelah fitur didapatkan maka akan dilakukan seleksi fitur menggunakan algoritma *Particle Swarm Optimization*. Pada algoritma PSO ini diperlukan beberapa masukan yang meliputi: jumlah iterasi, ukuran *swarm*, bobot *inersia* (w), koefisien akselerasi 1 (c_1), dan koefisien akselerasi 2 (c_2). Setelah itu dilakukan proses *Naïve Bayes* seperti standar deviasi, Gaussian,

likelihood dan *posterior* yang mana bertujuan untuk mendapatkan hasil klasifikasi *Naïve Bayes*. Hasil klasifikasi *Naïve Bayes* ini adalah yang terdapat pada *posterior* dengan cara memilih nilai yang terbesar disbanding yang lainnya setelah *posterior* didapatkan maka akan dibuatkan suatu tabel kebenaran yang berguna menghasilkan nilai *fitness* untuk algoritma PSO. Proses seleksi fitur algoritma PSO pada *Naïve Bayes* ini akan berhenti ketika telah mencapai jumlah partikel makksimal dan juga iterasi maksimal. Untuk contoh nilai-nilai fitur dari gambar mahasiswa yang telah dikonversikan ke HSV, YCbCr, RGB, HSI dan CIELab dapat dilihat pada Tabel 1.

Tabel 1. Data Ekstraksi Fitur Gambar Wajah

*Ta bel-1	*Ta bel-2	*Ta bel-3	*Ta bel-4	*Ta bel-5	*Ta bel-6	*Ta bel-7
1	H	22,1	22,5	19,5	21,9	Oran g Ke-1
	S	65,6	62,7	59,7	54,7	
	V	61,5	60	64,3	66,6	
	R	51,8	50,4	50	47,4	
	G	30,3	30,6	29,8	31,0	
	B	17,8	18,8	20,1	21,5	
	Y	107,1	106,8	114,0	124,7	
	Cb	98	99,8	100,8	101	
	Cr	163,5	160,9	163,6	160,2	
	H	18,1	18,3	18	18,4	
	S	46,5	43,5	39,6	35,4	
	I	39,6	39,6	42,8	46,7	
	L	45,7	45,5	48,3	52,3	
	A	23,2	21,1	23,7	19,9	
	B	41,7	38,8	38,5	35,9	
	C	38,4	40	35,6	33,3	
	M	63,9	63,5	61,5	56,4	
	Y	78,8	77,6	74,1	69,8	

***Keterangan Tabel:**

Tabel 1 = Data Ke-

Tabel 2 = Nama Fitur

Tabel 3 = Nilai Fitur Baris 1 Kolom 1

Tabel 4 = Nilai Fitur Baris 1 Kolom 2

Tabel 5 = Nilai Fitur Baris 2 Kolom 1

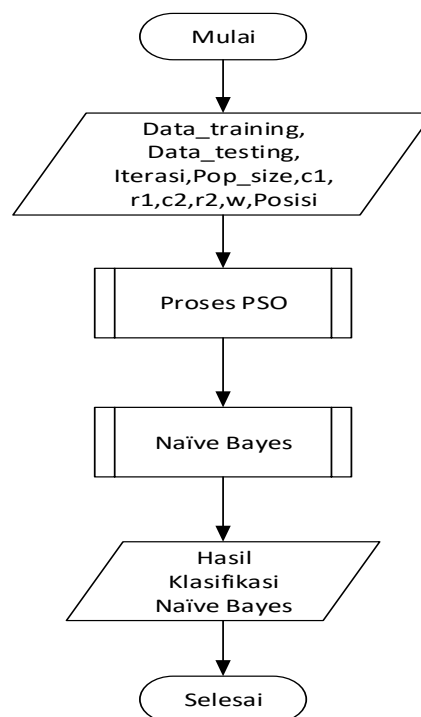
Tabel 6 = Nilai Fitur Baris 2 Kolom 2

Tabel 7 = Nama Kelas

3.2 Siklus Penyelesaian Masalah

Siklus algoritma PSO dan *Naïve Bayes* merupakan urutan penyelesaian masalah menggunakan algoritma *Particle Swarm Optimization* dan *Naïve Bayes* secara sekuensial. Optimasi menggunakan algoritma PSO bertujuan untuk memperoleh nilai yang optimum berdasarkan nilai *fitness* yang dihasilkan oleh

partikel sebagai representasi solusi seleksi fitur. Nilai densitas yang dihasilkan dapat memberikan nilai referensi terhadap proses algoritma *Naïve Bayes*. Diagram alir proses pengenalan wajah menggunakan PSO dan *Naïve Bayes* ditunjukkan pada Gambar 2.



Gambar 2. Diagram Alir Proses PSO dan *Naïve Bayes*

Pada algoritma PSO terdapat sejumlah partikel-partikel yang merupakan cikal bakal untuk melakukan proses seleksi fitur, dimana pada partikel tersebut akan dilakukan inisialisasi sebanyak jumlah fitur yang akan diproses, partikel-partikel tersebut akan direpresentasikan pada Tabel 2.

Tabel 2. Representasi Partikel PSO

Partikel ke-	Dimensi ke-	X1	X2	X3	X4	X5	X6	Xn
$x_1(0)$		1	1	0	1	1	0	0
$x_m(i)$		1	1	0	1	1	0	0

Keterangan Tabel :

$x_1 - x_n$ = Dimensi dari partikel diinisialisasi sejumlah fitur yang akan diproses sejumlah n .

$x_1(0)$ = Partikel ke-1 dan Iterasi ke 0.

$x_m(i)$ = Partikel ke- m dan Iterasi ke i .

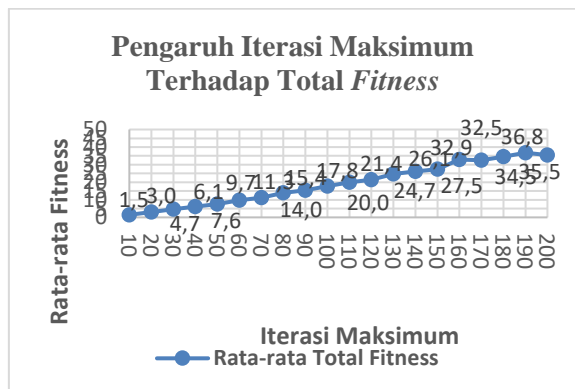
4. PENGUJIAN DAN ANALISIS

4.1 Hasil Pengujian dan Analisis Terhadap Iterasi Maksimum

Iterasi maksimum adalah pengujian dimana iterasi-iterasi tersebut untuk mengetahui nilai *fitness* terbaik. Iterasi maksimum yang akan diuji adalah pada kelipatan 10, yakni mulai dari iterasi 10 sampai dengan 200. Tiap percobaan iterasi tersebut akan dilakukan pengujian sebanyak 5 kali. Pada pengujian kali ini parameter-parameter yang diujikan meliputi:

Ukuran populasi = 4
 Koefisien akselerasi 1 (c_1) = 0,5
 Koefisien akselerasi 2 (c_2) = 0,5
 Bobot inersia (w) = 0,5

Pada hasil pengujian iterasi maksimum pada penelitian kali ini didapatkan hasil *fitness* yang berbeda. Selain itu kita juga dapat mengetahui rata-rata total *fitness* terbesar berada pada iterasi ke 190 yaitu 36,799. Dari pengujian iterasi maksimum ini dapat dilihat pada Gambar 3.



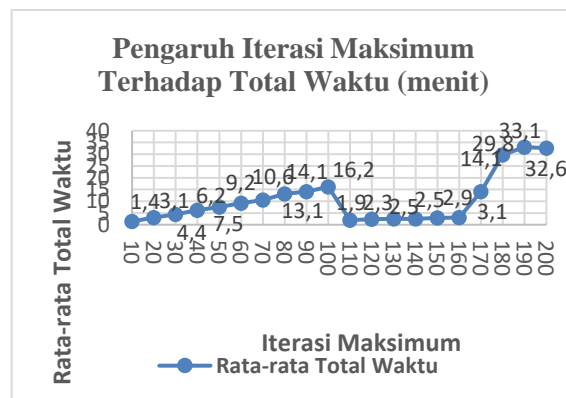
Gambar 3. Grafik Hasil Pengujian Iterasi Maksimum

Berdasarkan grafik pada Gambar 3, dapat diambil kesimpulan bahwa iterasi sangat mempengaruhi hasil *fitness* yang didapatkan tiap kali jumlah iterasinya. Hal ini dikarenakan sering terjadinya proses *update* posisi yang terjadi pada tiap iterasi. Dalam menentukan jumlah iterasi maksimum juga disarankan berdasarkan kompleksitas dari masalah yang akan diselesaikan. Jumlah iterasi apabila sangat sedikit ataupun terlalu banyak akan membuat komputasi semakin lama apabila hanya jumlah maksimum iterasi ini yang menjadi suatu patokan (Englebercth, 2007).

Dari grafik pada Gambar 3 juga dapat disimpulkan bahwa pengaruh iterasi terhadap

nilai *fitness* adalah belum terjadinya konvergen. Hal ini diakibatkan posisi selanjutnya yang akan ditempuh oleh suatu partikel akan dipengaruhi oleh posisi terbaik dari partikel yang ada pada sebelumnya dan juga posisi terbaik dari semua partikel. Apabila posisi terbaik tersebut tidak akan terjadi perubahan di *local optimum*, maka partikel akan cepat mengalami konvergen lalu sebaliknya apabila posisi terbaik tersebut masih adanya perubahan di *local optimum*, maka partikel belum dapat konvergen. Hal inilah yang terjadi pada pengujian iterasi penelitian kali ini belum terjadinya konvergen dikarenakan masih adanya perubahan di *local optimum*.

Hasil pengujian terhadap iterasi maksimum terhadap total waktu dapat dilihat pada Gambar 4.



Gambar 4. Grafik Hasil Waktu Pengujian Iterasi Maksimum

Dari grafik pada Gambar 4 yaitu pengaruh pengujian iterasi terhadap rata-rata total waktu adalah ada beberapa pengujian semakin naik iterasinya maka semakin naik pula total waktunya, namun ada juga yang naik iterasinya rata-rata total waktunya menurun hal ini diakibatkan oleh posisi fitur partikel = 0. Fitur pada posisi PSO tidak dieksekusi oleh *Naïve Bayes* untuk mendapatkan akurasi, sehingga tidak adanya perhitungan pada fitur pada posisi partikel = 0. Hal itu akan mempercepat jalannya eksekusi program selama pengujian jumlah iterasi.

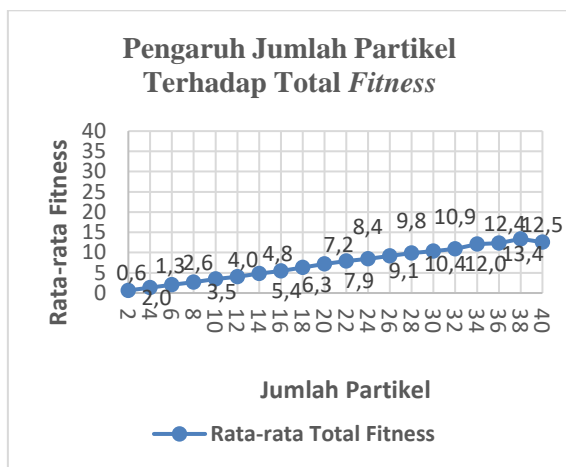
4.2 Hasil Pengujian dan Analisis Terhadap Jumlah Partikel

Pengujian jumlah partikel mempunyai tujuan agar mengetahui nilai *fitness* terbaik. Pada pengujian ini nantinya jumlah partikel yang akan diuji adalah partikel dengan kelipatan 2, mulai dari 2 sampai dengan 40, dengan masing-masing percobaan tiap jumlah partikelnya

sebanyak 5 kali. Pada pengujian kali ini parameter- parameter yang diujikan meliputi:

Iterasi maksimum = 10
 Koefisien akselerasi 1 (c_1) = 0,5
 Koefisien akselerasi 2 (c_2) = 0,5
 Bobot inersia (w) = 0,5

Pada hasil pengujian jumlah partikel didapatkan hasil *fitness* yang berbeda. Selain itu kita juga dapat mengetahui rata-rata total *fitness* terbesar berada pada jumlah partikel ke 38 yaitu 13,38. Dari pengujian jumlah partikel dapat dilihat pada Gambar 5.

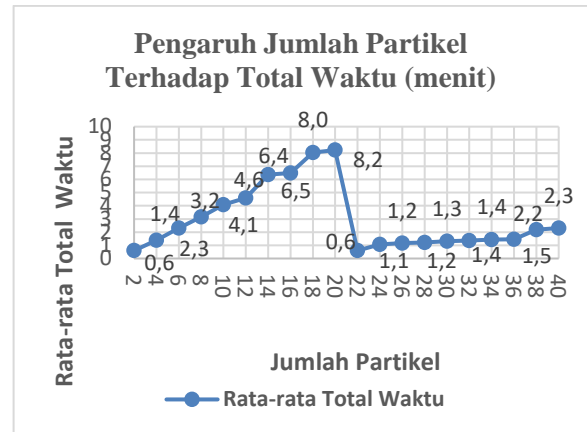


Gambar 5. Grafik Hasil Pengujian Jumlah Partikel

Berdasarkan grafik pada Gambar 5, dapat diambil kesimpulan bahwa apabila partikel semakin besar maka semakin baik pula solusi yang dihasilkan dikarenakan partikel yang banyak tersebut akan membuat banyak pilihan posisi, oleh karena itulah sangat besar kemungkinan yang akan didapatkan untuk mencapai solusi yang optimal. Selain itu ada pendapat yang mengatakan bahwa umumnya jumlah partikel 20 – 40 sudah cukup mampu dalam mendapatkan solusi yang optimal (Zerda, 2009).

Dari grafik pada Gambar 5 juga dapat disimpulkan bahwa pengaruh partikel terhadap nilai *fitness* adalah belum terjadinya konvergen. Hal ini diakibatkan oleh banyaknya partikel yang memiliki keanekaragaman dalam merepresentasikan solusi penyelesaian masalah. Sebuah partikel memiliki posisi dan kecepatan yang berbeda yang mana telah menentukan representasi solusi secara mandiri saat itu juga.

Hasil pengujian terhadap jumlah partikel terhadap total waktu dapat dilihat pada Gambar 6.



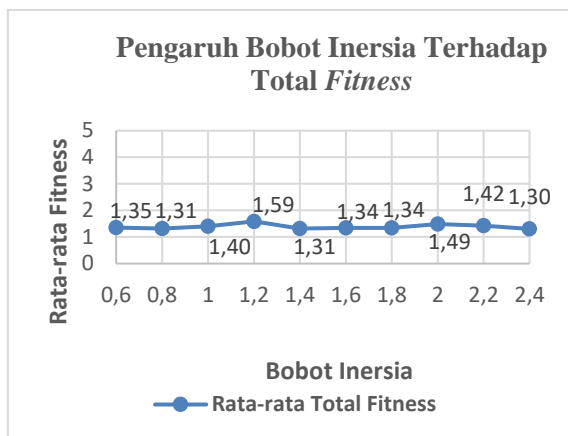
Gambar 6. Grafik Hasil Waktu Pengujian Jumlah Partikel

4.3 Hasil Pengujian dan Analisis Terhadap Bobot Inersia

Bobot inersia (w) merupakan pengujian akan mengetahui nilai *fitness* yang terbaik. Pada pengujian bobot inersia ini yang akan diuji ini nantinya adalah angka untuk masukan bobot inersia (w) dalam PSO dengan nilai 0,6 – 2,4. Setelah nilai bobot inersia dimasukkan maka akan dilakukan pengujian sebanyak 5 kali pada tiap nilai bobot inersianya. Pada pengujian kali ini parameter- parameter yang diujikan meliputi:

Iterasi maksimum = 10
 Jumlah partikel = 4
 Koefisien akselerasi 1 (c_1) = 0,5
 Koefisien akselerasi 2 (c_2) = 0,5

Pada hasil pengujian iterasi maksimum didapatkan hasil *fitness* yang berbeda. Selain itu kita juga dapat mengetahui rata-rata total *fitness* terbesar berada pada bobot inersia 1,2 yaitu 1,588. Setelah bobot inersia mencapai 1.2 tersebut hasil rata-rata *fitness* tidak mengalami kenaikan yang signifikan. Hal ini tidak seperti pada pengujian sebelumnya yaitu iterasi maksimum dan juga jumlah partikel yang mana semakin naik nilainya maka akan semakin tinggi nilai *fitnessnya*. Dari pengujian bobot inersia dapat dilihat pada Gambar 7.

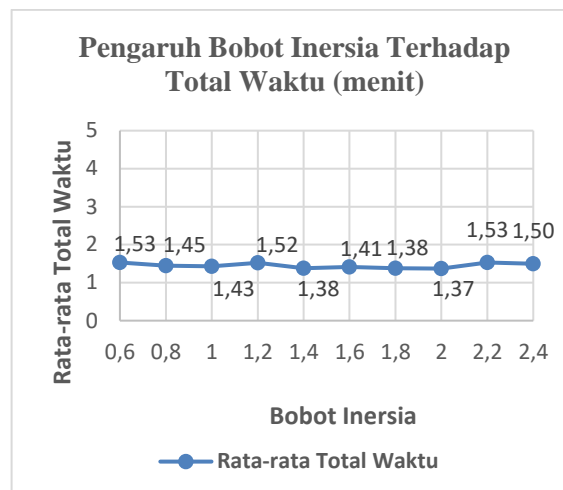


Gambar 7. Grafik Hasil Pengujian Bobot Inersia

Berdasarkan grafik pada Gambar 7, dapat diambil kesimpulan apabila bobot inersia semakin besar maka akan terjadi penurunan dari kecepatan tiap iterasinya, dengan kata lain semakin besar bobot inersia tersebut maka kecepatan partikel akan diperlambat pada titik awal pencarian solusi. Apabila kecepatan melambat pada awal titik pencarian solusi hal ini akan memberikan kesempatan terhadap eksploitasi lokal. Pada proses eksploitasi ini mempunyai kegunaan untuk pencarian solusi dalam suatu wilayah tertentu sebelum melanjutkan eksplorasi ke wilayah yang lain, namun ada beberapa hal yang perlu diperhatikan jika pada kesempatan tersebut eksploitasi tersebut sangatlah kecil, yang terjadi adalah partikel-partikel tersebut akan cenderung akan melakukan eksploitasi wilayah baru dan akan kehilangan kesempatan untuk mengeksplorasi pada wilayah yang sedang dikunjungi, hal inilah yang menjadi permasalahan karena solusi yang optimal dari wilayah itu akan dilewati begitu saja (Novitasari, 2015).

Dari grafik pada Gambar 7 juga dapat disimpulkan bahwa pengaruh bobot inersia terhadap nilai *fitness* adalah terjadinya konvergensi dini. Hal ini diakibatkan oleh fungsi bobot inersia itu sendiri yang hanya sebagai pengontrol dampak perubahan dari kecepatan yang diberikan oleh partikel. Perubahan dari kecepatan yang diberikan oleh partikel pada saat pengujian cukup baik sehingga menghasilkan hasil solusi yang cukup optimal. Pada pengujian bobot inersia ini terjadi hanya satu kali kenaikan yang berarti yaitu pada nilai bobot inersia 1,2.

Hasil pengujian bobot inersia terhadap total waktu dapat dilihat pada Gambar 8.



Gambar 8. Grafik Hasil Waktu Pengujian Bobot Inersia

4.4 Hasil Pengujian dan Analisis Global

Pada pengujian jumlah iterasi didapatkan kesimpulan bahwa iterasi sangat mempengaruhi hasil *fitness* yang didapatkan tiap kali jumlah iterasinya. Hal ini dikarenakan sering terjadinya proses *update* posisi yang terjadi pada tiap iterasi. Dalam menentukan jumlah iterasi maksimum juga disarankan berdasarkan kompleksitas dari masalah yang akan diselesaikan. Jumlah iterasi apabila sangat sedikit ataupun terlalu banyak akan membuat komputasi semakin lama apabila hanya jumlah maksimum iterasi ini yang menjadi suatu patokan (Englebercth, 2007). Hasil pengaruh iterasi terhadap nilai *fitness* pada penelitian kali ini adalah belum terjadinya konvergen. Hal ini diakibatkan posisi selanjutnya yang akan ditempuh oleh suatu partikel akan dipengaruhi oleh posisi terbaik dari partikel yang ada pada sebelumnya dan juga posisi terbaik dari semua partikel. Apabila posisi terbaik tersebut tidak akan terjadi perubahan di *local optimum*, maka partikel akan cepat mengalami konvergen lalu sebaliknya apabila posisi terbaik tersebut masih adanya perubahan di *local optimum*, maka partikel belum dapat konvergen. Hal inilah yang terjadi pada pengujian iterasi penelitian kali ini belum terjadinya konvergen dikarenakan masih adanya perubahan di *local optimum*.

Pada pengujian jumlah partikel didapatkan kesimpulan bahwa apabila partikel semakin besar maka semakin baik pula solusi yang dihasilkan dikarenakan partikel yang banyak tersebut akan membuat banyak pilihan posisi partikel, oleh karena itulah sangat besar kemungkinan yang akan didapatkan untuk mencapai solusi yang optimal. Selain itu ada

pendapat yang mengatakan bahwa umumnya jumlah partikel 20 – 40 sudah cukup mampu dalam mendapatkan solusi yang optimal (Zerda, 2009). Hasil pengaruh partikel terhadap nilai *fitness* adalah belum terjadinya konvergen. Hal ini diakibatkan oleh banyaknya partikel yang memiliki keanekaragaman dalam merepresentasikan solusi penyelesaian masalah. Sebuah partikel memiliki posisi dan kecepatan yang berbeda yang mana telah menentukan representasi solusi secara mandiri saat itu juga.

Pada pengujian bobot inersia didapatkan kesimpulan apabila bobot inersia semakin besar maka akan terjadi penurunan dari kecepatan tiap iterasinya, dengan kata lain semakin besar bobot inersia tersebut maka kecepatan partikel akan diperlambat pada titik awal pencarian solusi. Apabila kecepatan melambat pada awal titik pencarian solusi hal ini akan memberikan kesempatan terhadap eksploitasi lokal. Pada proses eksploitasi ini mempunyai kegunaan untuk pencarian solusi dalam suatu wilayah tertentu sebelum melanjutkan eksplorasi ke wilayah yang lain, namun ada beberapa hal yang perlu diperhatikan jika pada kesempatan tersebut eksploitasi yang terjadi sangatlah kecil, yang terjadi adalah partikel-partikel tersebut akan cenderung akan melakukan eksploitasi wilayah baru dan akan kehilangan kesempatan untuk mengeksplorasi pada wilayah yang sedang dikunjungi, hal inilah yang menjadi permasalahan karena solusi yang optimal dari wilayah itu akan dilewati begitu saja (Novitasari, 2015). Hasil pengaruh bobot inersia terhadap nilai *fitness* pada penelitian kali ini adalah terjadinya konvergensi dini. Hal ini diakibatkan oleh fungsi bobot inersia itu sendiri yang hanya sebagai pengontrol dampak perubahan dari kecepatan yang diberikan oleh partikel. Perubahan dari kecepatan yang diberikan oleh partikel pada saat pengujian cukup baik sehingga menghasilkan hasil solusi yang cukup optimal.

5. KESIMPULAN

Setelah dilakukan penelitian dan juga pengujian diatas didapatkan kesimpulan sebagai berikut:

1. Implementasi algoritma *Particle Swarm Optimization* dalam penyeleksian fitur pada pengenalan pola wajah dapat berjalan cukup baik dengan cara sebagai berikut:
 - a. Inisialisasi partikel dimana pada posisi partikel awal ini didefinisikan secara

acak dan menghasilkan partikel-partikel untuk dilakukannya seleksi fitur.

- b. Lalu perhitungan *fitness* didapatkan dari hasil akurasi klasifikasi algoritma *Naïve Bayes* berdasarkan tabel kebenaran.
 - c. Kemudian setelah itu *update* kecepatan yang sangat berguna pada perhitungan untuk melakukan perpindahan posisi pada saat proses sedang terjadi.
 - d. Lalu dilakukan *update* posisi untuk menentukan posisi partikel setelah dilakukannya perpindahan.
 - e. Kemudian menentukan *Pbest* untuk evaluasi serta membandingkan hasil dari tiap *update* posisi.
 - f. Terakhir adalah menentukan *Gbest* yang mana berfungsi untuk mencari solusi terbaik secara global pada partikel dengan posisi yang terbaik.
2. Implementasi algoritma *Naïve Bayes* dalam proses klasifikasi wajah dapat berjalan dengan baik, melalui tahapan sebagai berikut:
 - a. Menghitung mean berguna mencari rata-rata, varian, standar deviasi, Gaussian, *Likelihood*, dan juga Posterior.
 - b. Pada *Naïve Bayes* ini terjadi seleksi fitur yang mana untuk seleksi fitur tersebut membutuhkan posisi partikel dari PSO. Dapat disimpulkan bahwa *Naïve Bayes* ini dapat berjalan setelah algoritma PSO memberikan posisi partikel dalam proses klasifikasinya, setelah posisi partikel dari PSO didapatkan dilakukan proses klasifikasi.
 - c. Hasil klasifikasinya dapat menentukan suatu tabel kebenaran/akurasi yang nantinya akan menjadi nilai *fitness* untuk tiap partikel PSO.
3. Pada pengujian pengaruh iterasi didapatkan kesimpulan bahwa iterasi dengan total nilai *fitness* terbaik pada jumlah partikel ke 38 yaitu 13,38, kemudian pada pengujian pengaruh jumlah iterasi didapatkan kesimpulan bahwa total *fitness* terbesar berada pada iterasi 190 yaitu 36,799, dengan kata lain semakin besar iterasi maka *fitness* semakin baik pula dan pengujian yang terakhir yaitu pada bobot inersia berada, *fitness* tertinggi pada bobot inersia 1,2 yaitu 1,588.

DAFTAR PUSTAKA

- Al Fatta H., 2009. Rekayasa Sistem Pengenalan Wajah: Membangun Sistem Presensi Karyawan menggunakan Microsoft Visual Basic 6.0 dan Microsoft Access. Andi. Yogyakarta.
- Bustami., 2013. Penerapan Algoritme *Naive Bayes* Untuk Mengklasifikasi Data Nasabah Asuransi, *TECHSI : Jurnal Penelitian Teknik Informatika*.
- Brainard D.H., 1989. *Calibration of a computer controlled color monitor, Color Research & Application*.
- Choi Y.J., Ro M.Y., Plataniotis N.K., 2011. *Boosting Color Feature Selection for Color Face Recognition. Accepted Manuscript for Publication in IEEE Transaction on Image Processing*.
- Cholissodin I., Riyandani E., 2016. *Swarm Intelligence*. Fakultas Ilmu Komputer. Universitas Brawijaya. Malang.
- Esti N.K dan Yamasari Y., 2013. Aplikasi Diagnosa Kanker Kandungan Dengan Menggunakan Metode *Naive Bayes*. Jurnal: Manajemen Informatika Fakultas Teknik, Universitas Negeri Surabaya.
- Eberhart R.C dan Shi Y., 2011. *Particle Swarm Optimization : Developmetns, Applications and Resources. Proceedings of Congress on evolutionary computation*.
- Englebercth A.P., 2007. *Computational Intelligent : An Introduction 2nd ed., West Sussex : John Willwy & Sons Ltd*.
- Foka P dan Ramji D.P., 2002. *CCAAT/Enhancer-Binding Proteins: Structure, Function and Regulation. Biochemical Journal*.
- Ford A dan Roberts A., 1998. *Color Space Conversions*.
- Fais A., Aditya D.M., Mulya S., Ramadien D., Sani A., 2014. Klasifikasi Calon Pendorong Darah Dengan Metode *Naive Bayes Classifier*. S1. Universitas Brawijaya. Malang.
- Jain A.K., Duin R.P.W., Mao J., 2000. *Statistical pattern recognition: A review Transaction on Pattern Analysis and Machine Intelligence*. IEEE.
- Jackson R., MacDonald L., Freeman K., John W & Sons., 1994. *Computer Generated Colour*.
- Kusumadewi S., 2003. *Artificial Intelligence (Teknik dan Aplikasinya)*. Graha Ilmu .Yogyakarta.
- Khusna R.A., 2016. Implementasi Algoritma *Particle Swarm Optimization* Untuk Optimasi Pemerataan Guru Mata Pelajaran Di Kabupaten Lamongan. S1. Universitas Brawijaya. Malang.
- Karina N.E dan Yamasari Y., 2013. Aplikasi Diagnosa Kanker Kandungan dengan Menggunakan Metode *Naive Bayes* (Studi Kasus: Rumah Sakit Islam Surabaya). Jurnal Manajemen Informatika.
- Novitasari D., 2015. Optimasi *Support Vector Regression* dengan *Particle Swarm Optimization* untuk *Software Effort Estimasi*. S1. Universitas Brawijaya. Malang.
- Permana K.E dan Hashim S.Z.M., 2010. *Fuzzy Membership Function Generation using Particle Swarm Optimization. International Journal Problems Compt*.
- Park T.S., Lee J.H., Choi B., 2009. *Optimization for Artificial Neural Network with Adaptive inertial weight of particle swarm optimization. IEEE International Conference on Cognitive Informatics*.
- Putra T.W.A., 2013. Pengenalan Wajah Dengan Matriks Kookurensi Aras Keabuan Dan Jaringan Syaraf Tiruan Probabilistik. S2. Universitas Diponegoro. Semarang.
- Ramadhan R.M., Rehab F., Abdel K., 2009. *Face Recognition Using Particle Swarm Optimization-Based Selected Features. International Journal of Signal Processing, Image Processing and Pattern*.
- Saleh A., 2015. Implementasi Metode Klasifikasi *Naive Bayes* Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga. *Creative Information Technology Journal* (Citec).
- Santosa B., Willy P., 2011. Metode Metaheuristik, Konsep dan Implementasi. Graha Ilmu. Surabaya.
- Sutanto T., 2010. Kombinasi Penyaring Warna Kulit Kplisit Pada Bidang Warna RGB

Dan YCBCR Untuk Meningkatkan Akurasi Sistem Pendeteksian Warna Kulit. STIKOM. Surabaya.

Sproson W.N., Adam H Ltd., 1983. *Colour Science in Television and Display Systems*.

Tuegeh M., Soeprijanto., Purnomo., Mauridhi H., 2009. *Modified Improved Particle Swarm Optimization For Optimal Generator Scheduling*. Seminar Nasional Aplikasi Teknologi Informasi. Yogyakarta.

Xue B.X., Ma X., Gu J., Li Y.B., 2013. *An improved extremelearning machine based on variable-length particle swarm optimization, Proc. 10th IEEE International Conference on Robotics and Biomimetics*. IEEE Computer Society.

Yu Y., 2015. *Study On Student Classification Management Based On Bayes Network. International Journal of Simulation Systems, Science & Technology*.

Zerda E.R., 2009. Analisis dan penerapan Algoritme *Particle Swarm Optimization* (PSO) pada Optimasi Penjadwalan Sumber Daya Proyek. S1. Institut Teknologi Telkom. Bandung.