

BAB 3

**ALGORITMA PENGENALAN SUDUT FOTO DENGAN DATA
RATA-RATA MASING-MASING SUDUT SERTA DATA FUZZY
SEGITIGA MASING-MASING SUDUT**

Bab ini akan membahas algoritma-algoritma yang dikembangkan dengan dasar penelitian sebelum ini. Penelitian yang dilakukan dibagi menjadi dua kelompok besar, yaitu penelitian dengan menggunakan data rata-rata masing masing sudut, serta penelitian dengan menggunakan data *fuzzy* segitiga (nilai minimal, nilai rata-rata, dan nilai maksimal).

3.1 Sistem dengan Masukan Data Rata-Rata per Sudut sebagai Data Acuan Awal dan Data Uji

Sistem ini berbasis kepada penelitian sebelumnya (Sanabila, 2008), data acuan awal yang jarak sudutnya besar kemudian dijadikan titik-titik yang akan diinterpolasi sehingga menghasilkan data-data acuan baru yang jarak sudutnya lebih kecil. Selanjutnya data uji dibandingkan jarak Euclid-nya dengan semua data acuan yang sesuai lalu dicari data acuan dengan jarak yang paling dekat sebagai tebakan dari implementasi algoritma (Sanabila, 2008).

Mengacu kepada persoalan-persoalan yang ditemukan pada algoritma di penelitian sebelumnya (Sanabila, 2008) yang dijelaskan pada akhir Bab 2, maka dibuatlah modifikasi-modifikasi seperti berikut:

1. Pada tahap ekstraksi data di penelitian sebelumnya (Sanabila, 2008), merujuk pada Gambar 2.10, data dikelompokkan terhadap sudut vertikal, sudut horizontal, dan individu pada citra (atau disebut kelas wajah), sedangkan pada penelitian ini data dikelompokkan hanya per sudut. Selanjutnya data masing-masing sudut tersebut dicari rata-ratanya yang kemudian menjadi data acuan. Modifikasi tersebut akan dibahas pada Subbab 3.1.1.

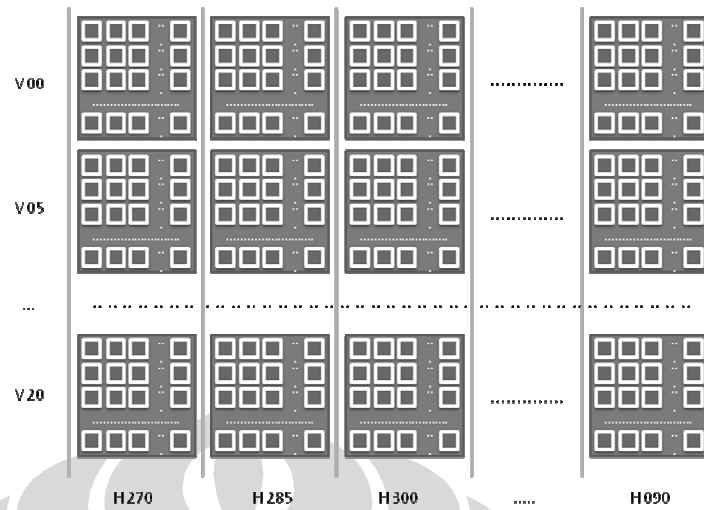
2. Pada PCA yang dilakukan, pada penelitian sebelum ini data uji yang akan ditransformasi ke ruang Eigen tidak dinormalisasi terlebih dahulu sebelum dikalikan dengan matriks transformasi, sedangkan pada penelitian ini normalisasi pada data uji dilakukan dengan sedikit modifikasi pada algoritma PCA. Modifikasi tersebut akan dibahas lebih lanjut di Subbab 3.1.2.
3. Pada tahap pembuatan data acuan baru menggunakan interpolasi Bezier kuadrat, salah satu kelemahan dari metode interpolasi tersebut coba diatasi dengan menggunakan metode penempatan titik kontrol kedua yang tadinya merupakan titik data juga sehingga titik data tersebut tidak lagi menjadi titik kontrol dan tidak lagi tidak dilewati oleh garis hasil interpolasi. Modifikasi tersebut akan dibahas pada Subbab 3.1.3.

Selain modifikasi pada algoritma, ada beberapa perbedaan dalam skema eksperimen yang dilakukan, diantaranya hasil eksperimen yang tidak menggunakan PCA dalam tahap ekstraksi datanya juga ditampilkan.

3.1.1 Pengelompokan Data per Sudut dan Pengambilan Rata-Rata Masing-Masing Sudut sebagai Data Acuan Awal

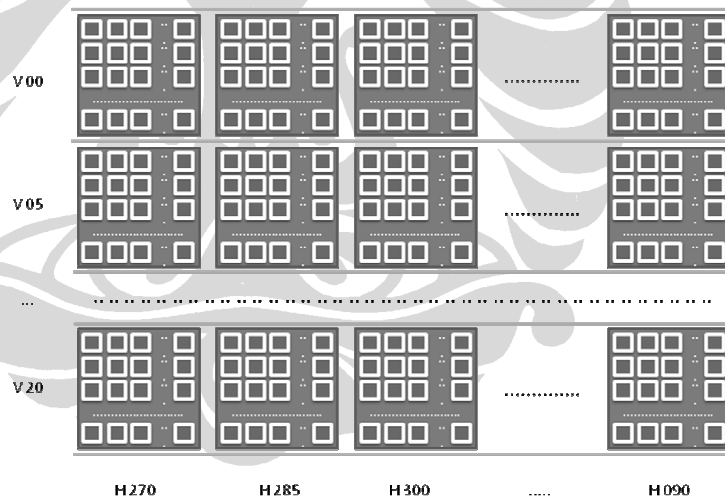
Pada penelitian sebelumnya (Sanabila, 2008), data dikelompokkan terhadap sudut vertikal, sudut horizontal, dan individu pada citra serta menjadikan masing-masing citra sebagai data acuan awal. Penjelasan lebih rincinya dapat dilihat pada Bab 2 serta pada Laporan TA: Penentuan Sudut Pandang Wajah 3 Dimensi dengan Menggunakan Interpolasi Linier dan Interpolasi Spline sebagai Fungsi Pembentuk Garis Ciri (Sanabila, 2008).

Dalam penelitian ini, data dikelompokkan hanya terhadap sudut vertikalnya dan sudut horizontalnya. Gambar 3.1 mendeskripsikan skema pengelompokan data berdasarkan sudut horizontal yang dilakukan.



Gambar 3.1 Pengelompokan Data Berdasarkan Sudut Horizontal

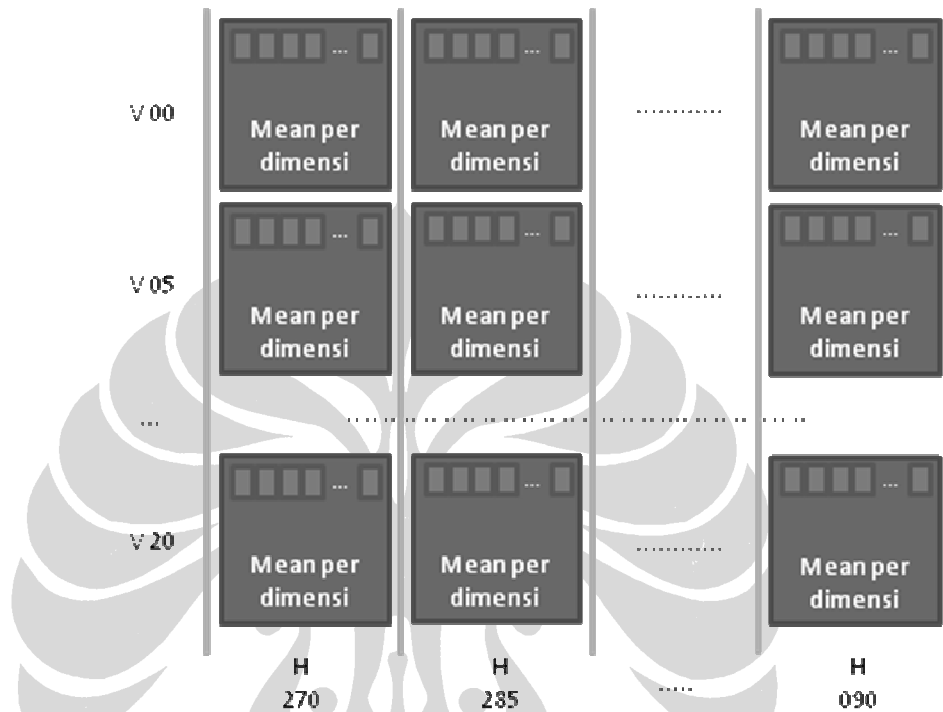
Sementara itu Gambar 3.2 adalah penggambaran pengelompokan data berdasarkan sudut vertikal.



Gambar 3.2 Pengelompokan Data Berdasarkan Sudut Vertikal

Pada Gambar 3.1 dan Gambar 3.2 terlihat bahwa data hanya dipisahkan berdasarkan sudutnya, sementara data-data dengan berbagai variasi orang dan variasi pose dikumpulkan berdasarkan sudutnya.

Selanjutnya data yang akan digunakan sebagai data acuan awal merupakan data rata-rata masing masing sudut. Pada akhirnya skema pengelompokan berdasarkan sudut horizontal akan menjadi seperti Gambar 3.3.



Gambar 3.3 Data yang Diambil Hanya Rata-Rata

Pada Gambar 3.3 terlihat bahwa data data dengan sudut yang sama sudah diekstrak rata-ratanya saja. Demikian pula yang berlaku saat pengelompokan berdasarkan sudut vertikal.

Alasan dari skema pengelompokan dibuat seperti itu adalah algoritma tidak bertujuan mengenali individu pada citra melainkan hanya sudut fotonya saja sehingga sebuah citra dimiliki oleh individu manapun seharusnya tidak berpengaruh.

3.1.2 Normalisasi Data Uji pada Transformasi Data Uji ke Ruang Eigen

Dalam pengembangan algoritma, pada saat pengujian, data uji dapat dimasukkan ke dalam sistem menjadi satu set data, namun pada keadaan sebenarnya data uji merupakan data tunggal yang tidak dapat dikaitkan dengan data uji lainnya

$$z_{du} = \frac{x_{du} - \bar{x}_{da}}{\sigma_{da}} \quad (3.1)$$

z_{du} : nilai *z-score* data uji.

x_{du} : nilai mentah data uji.

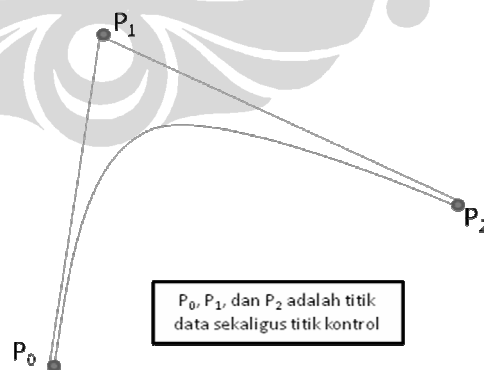
\bar{x}_{da} : nilai rata-rata data acuan yang sesuai.

σ_{da} : standar deviasi data acuan yang sesuai.

3.1.3 Penentuan Posisi Titik Kontrol Kedua pada Metode Interpolasi Bezier Kuadratik

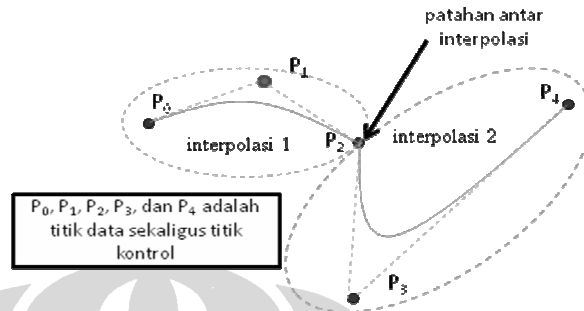
Metode interpolasi Bezier kuadratik memiliki dua kelemahan utama, yaitu:

1. Pembuatan interpolasi membutuhkan tiga titik kontrol. Titik kontrol yang dilewati garis interpolasi hanya titik kontrol pertama dan ketiga (awal dan akhir), sementara titik kontrol kedua tidak dilewati oleh interpolasi. Jika ketiga titik kontrol tersebut merupakan titik data, maka titik data kedua tidak akan dilewati oleh interpolasi. Gambaran dari kondisi tersebut terdapat pada Gambar 3.5.



Gambar 3.5 Titik Data Kedua Berperan Juga Sebagai Titik Kontrol Kedua

2. Jika memerlukan dua interpolasi, yaitu saat jumlah titik data melebihi tiga buah, sambungan antara interpolasi patah. Gambaran dari sambungan antara dua interpolasi terdapat pada Gambar 3.6.



Gambar 3.6 Patahan pada Sambungan antara Interpolasi

Untuk mengatasi masalah pada poin pertama, titik data kedua tidak dianggap sebagai titik kontrol kedua. Digunakanlah metode yang disebut Penempatan Titik Kontrol Kedua atau *Control Point Placement*.

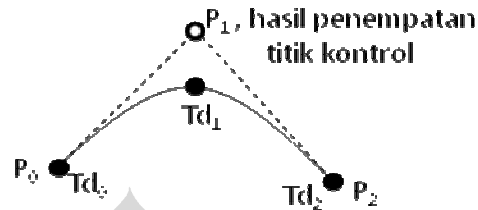
Misalkan ada tiga titik data (Td_0 , Td_1 , dan Td_2), Td_0 dan Td_2 dianggap titik kontrol awal dan akhir (P_0 dan P_2). Kemudian merujuk kepada persamaan 2.2, titik data Td_1 dianggap sebagai titik hasil interpolasi pada saat variabel t berada pada nilai tertentu (P_t). Maka dengan membalik persamaan 2.2, P_1 dapat ditentukan posisinya.

$$P_t = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2$$

$$\rightarrow P_1 = \frac{P_t - ((1-t)^2 P_0 + t^2 P_2)}{2t(1-t)} \quad (3.2)$$

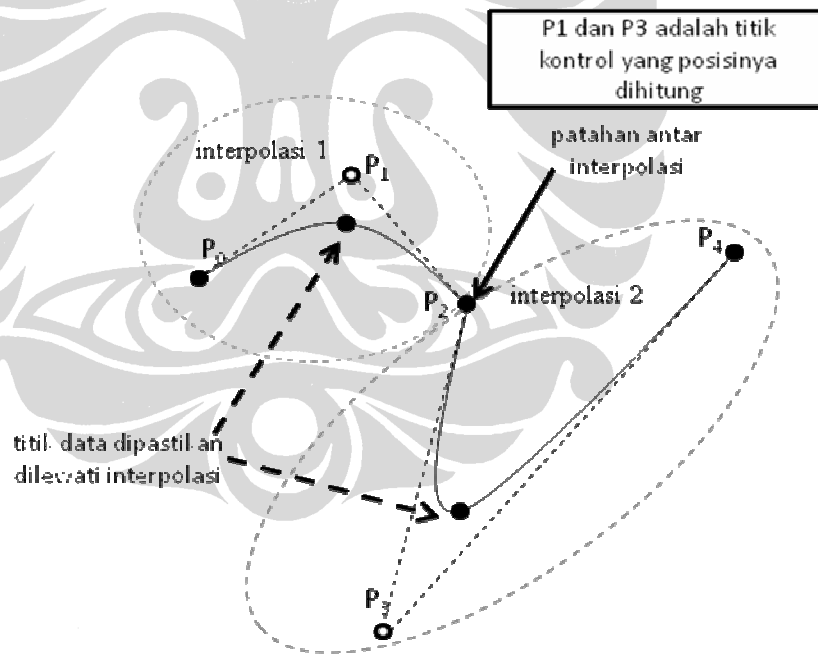
Posisi P_0 , P_2 , dan Td_1 (atau P_t pada persamaan tersebut) sudah diketahui, sementara nilai variabel t harus ditentukan sendiri. Dalam eksperimen yang dilakukan nilai variabel t ditentukan pada angka 0,5.

Dengan menggunakan metode tersebut, titik data kedua akan pasti akan dilewati oleh interpolasi pada $t = 0,5$. Gambar 3.7 mendeskripsikan interpolasi menggunakan metode tersebut.



Gambar 3.7 Interpolasi Bezier Kuadratik dengan Penempatan Titik Kontrol Kedua

Metode penempatan titik kontrol ini tidak dapat mengatasi patahnya sambungan antar interpolasi. Pada akhirnya interpolasi yang akan terbentuk akan menjadi seperti Gambar 3.8.



Gambar 3.8 Sambungan antar Interpolasi yang Masih Patah

Terlihat dari Gambar 3.8 tersebut bahwa patahan antar interpolasi masih terjadi, namun titik data tengah dapat dipastikan terlewati garis hasil interpolasi.

Pada eksperimen yang dilakukan (hasilnya dapat dilihat pada Bab 4), penerapan interpolasi Bezier kuadratik tanpa *Control Point Placement* tetap dilakukan. Hasil tersebut akan dibandingkan dengan yang menggunakan *Control Point Placement* untuk membuktikan apakah memang penggunaan *Control Point Placement* dapat memberikan hasil yang lebih baik.

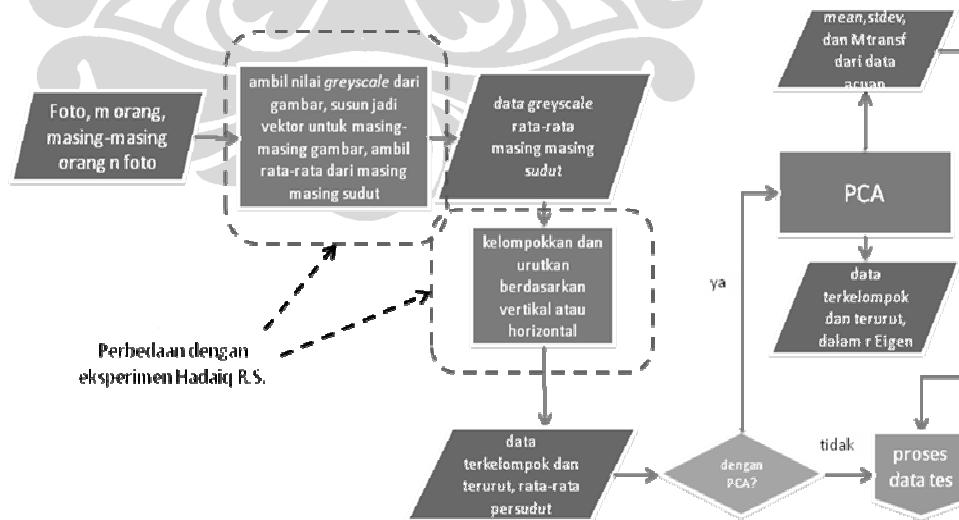
Dengan berbagai perubahan yang dilakukan, maka algoritma untuk sistem juga berubah. Secara garis besar algoritma yang dikembangkan dibagi menjadi tiga bagian, sama seperti algoritma sebelumnya, yaitu:

1. Ekstraksi data.
2. Pembuatan data acuan baru dari data acuan awal dengan menggunakan interpolasi.
3. Penentuan tebakan dan kalkulasi tingkat pengenalan.

Hasil akhir dari implementasi algoritma adalah hasil tebakan implementasi algoritma dan tingkat pengenalan.

3.1.4 Ekstraksi Data

Ekstraksi data dapat dibagi menjadi dua bagian, yaitu ekstraksi data acuan dan ekstraksi data uji. Gambar 3.9 adalah diagram alur ekstraksi data acuan.

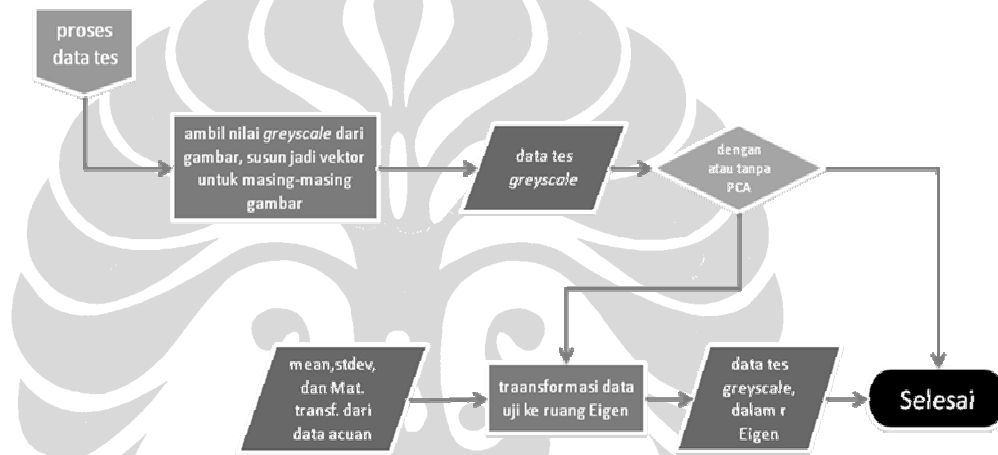


Gambar 3.9 Diagram Alur Ekstraksi Data Acuan dan Perubahan dari Sistem Sebelumnya

Algoritma tersebut tidak jauh berbeda dengan langkah-langkah ekstraksi data pada yang telah dijelaskan pada Bab 2. Perbedaan-perbedaan tersebut seperti yang ditandai pada gambar, yaitu:

1. Data rata-rata masing masing dimensi pada masing-masing sudut yang dijadikan data acuan awal.
2. Data hanya dikelompokkan berdasarkan sudut vertikal dan horizontalnya.

Pada Gambar 3.9, proses dilanjutkan dengan pemrosesan data uji. Gambar 3.10 adalah diagram alur proses ekstraksi data uji.



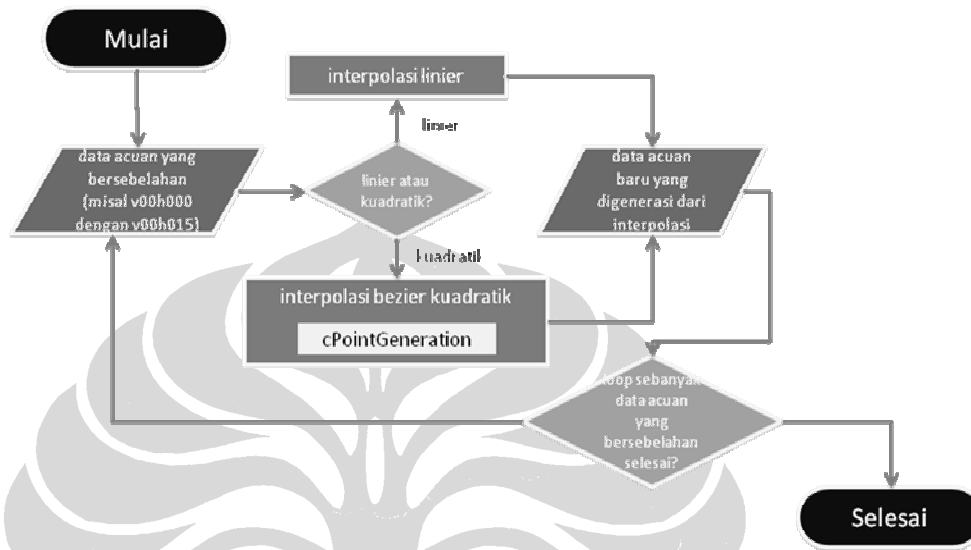
Gambar 3.10 Diagram Alur Ekstraksi Data Uji

Proses tersebut hampir tidak berbeda dengan yang dilakukan di Penentuan Sudut Pandang Wajah 3 Dimensi dengan Menggunakan Interpolasi Linier dan Interpolasi Spline sebagai Fungsi Pembentuk Garis Ciri (Sanabila, 2008), perbedaannya hanya terletak pada transformasi data uji ke ruang Eigen, yaitu seperti yang sudah dijelaskan pada Subbab 3.1.2.

3.1.5 Pembuatan Data Acuan Baru dari Data Acuan Awal dengan Menggunakan Interpolasi

Gambar 3.11 merupakan diagram alur untuk proses pembuatan data acuan baru dari data acuan awal dengan menggunakan interpolasi. Seperti yang sudah dijelaskan pada Subbab 3.1.3, perubahan yang dilakukan pada tahap ini adalah hanya penggunaan metode penempatan titik kontrol kedua saat menggunakan

interpolasi Bezier kuadrat. Selanjutnya algoritma penggunaan interpolasi yang dilakukan tidak berbeda dengan yang dilakukan pada penelitian sebelumnya (Sanabila, 2008).



Gambar 3.11 Diagram Alur Pembentukan Titik Ciri Baru

Dalam penggunaan interpolasi Bezier kuadrat, pada setiap tiga titik interpolasi dibuat. Sebelum membuat garis interpolasi tersebut, titik kontrol kedua digenerasi terlebih dahulu, baru interpolasi dibuat melewati titik kontrol kedua tersebut.

3.1.6 Penentuan tebakan dan kalkulasi tingkat pengenalan.

Secara umum hampir tidak ada perbedaan antara penelitian ini dengan Penentuan Sudut Pandang Wajah 3 Dimensi dengan Menggunakan Interpolasi Linier dan Interpolasi Spline sebagai Fungsi Pembentuk Garis Ciri (Sanabila, 2008) dalam penentuan hasil tebakan dan penghitungan tingkat pengenalan atau tingkat ketepatan tebakan. Tetapi ada dua perbedaan, yaitu:

1. Dilakukan pembulatan terhadap hasil tebakan ke kelipatan lima terdekat. Misalkan, hasil tebakan dari sistem untuk sudut horizontal adalah 333 derajat, maka hasil tebakannya akan dibulatkan menjadi 335, jika hasil tebakan sistem 332, maka akan dibulatkan menjadi 330. Hal ini dilakukan dengan alasan data tes yang memang memiliki sudut kelipatan lima serta kebutuhan sistem

pengenalan wajah 3 dimensi yang hanya membutuhkan masukan sudut dengan kelipatan 5 derajat saja.

2. Adanya toleransi lima derajat untuk sebuah tebakan dianggap benar atau salah. Misalkan jika tebakan sistem 335 tetapi sudut sebenarnya adalah 340, maka tebakan sistem tetap dianggap benar. Hal ini dilakukan karena pada kenyataannya belum ada sistem pengenalan yang bisa benar-benar akurat, juga pada kenyataannya foto dengan sudut 335 derajat dan 340 tidak terlalu terlihat berbeda, selain itu hal ini merupakan antisipasi jika dalam pengurutan jarak Euclid, hasil tertinggi kedua (atau lebih) yang masih dalam jangkauan *threshold* ternyata merupakan hasil yang seharusnya.

Selebihnya algoritma penentuan hasil tebakan dan kalkulasi tingkat ketepatan tebakan sistem yang dilakukan di penelitian ini tidak berbeda dengan yang dilakukan pada Penentuan Sudut Pandang Wajah 3 Dimensi dengan Menggunakan Interpolasi Linier dan Interpolasi Spline sebagai Fungsi Pembentuk Garis Ciri (Sanabila, 2008) yang telah dijelaskan di Bab 2. Skema eksperimen dan hasil eksperimen dari algoritma yang sudah dijelaskan akan dipaparkan pada Bab 4.

3.2 Sistem dengan Masukan Data Minimal, Rata-Rata, dan Maksimal (Triangular Fuzzy) per Sudut sebagai Data Acuan Awal dan Data Uji

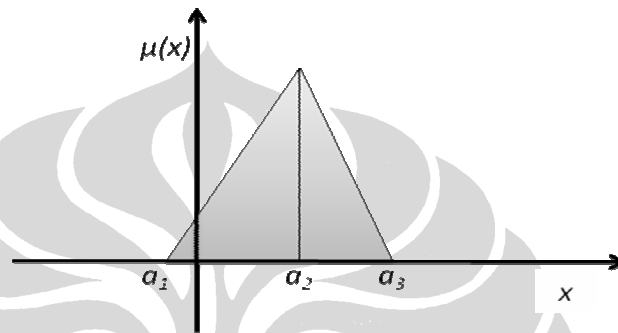
Sistem ini pada umumnya menggunakan metode yang hampir sama dengan sistem yang menggunakan data rata-rata. Perbedaan utama dari sistem ini adalah menggunakan data minimal, rata-rata, dan maksimal (*triangular fuzzy*) masing-masing dimensi per sudut untuk membentuk data acuan awal dan data acuan baru, serta data uji memiliki bentuk *fuzzy* segitiga dengan nilai minimal dan maksimal yang direkayasa.

3.2.1 Triangular Fuzzy Number

Triangular fuzzy number atau nilai *fuzzy* segitiga merupakan sebuah bilangan yang merepresentasikan distribusi satu set data. Bilangan ini dinyatakan dengan tiga angka berikut:

$$\bar{a} = (a^{(1)}, a^{(2)}, a^{(3)})$$

Nilai \bar{a} merupakan nilai *fuzzy*, a_1 merupakan nilai minimal, a_2 merupakan nilai rata-rata, dan a_3 merupakan nilai maksimal. Bentuk bilangan *fuzzy* segitiga dapat digambarkan seperti Gambar 3.12.



Gambar 3.12 Bentuk Bilangan *Fuzzy* Segitiga

Nilai keanggotaan ($\mu(x)$) berkisar antara 0 sampai 1. Definisi dari $\mu(x)$ itu sendiri dalam bilangan *fuzzy* segitiga adalah sebagai persamaan 3.3.

$$\mu(x) = \begin{cases} 0, & x < a^{(1)} \text{ atau } x > a^{(3)} \\ \frac{x - a^{(1)}}{a^{(2)} - a^{(1)}}, & a^{(1)} \leq x \leq a^{(2)} \\ \frac{a^{(3)} - x}{a^{(3)} - a^{(2)}}, & a^{(2)} \leq x \leq a^{(3)} \end{cases} \quad (3.3)$$

Berdasarkan persamaan 3.3 dapat diterjemahkan bahwa jika nilai x semakin mendekati nilai a_2 , maka nilai $\mu(x)$ akan semakin mendekati 1 (Kwang H. Lee, 2005).

3.2.2 Operasi Aritmetika pada Bilangan *Fuzzy* Segitiga

- Penjumlahan, definisinya seperti berikut (Denceux dan Masson, 2004):

$$\tilde{a} + \tilde{b} = (a^{(1)} + b^{(1)}, a^{(2)} + b^{(2)}, a^{(3)} + b^{(3)}) \quad (3.4)$$

- Pengurangan, definisinya seperti berikut (Denceux dan Masson, 2004):

$$\tilde{a} - \tilde{b} = (a^{(1)} - b^{(3)}, a^{(2)} - b^{(2)}, a^{(3)} - b^{(1)}) \quad (3.5)$$

- Perkalian dengan skalar, definisinya seperti berikut (Denceux dan Masson, 2004):

$$n \cdot \tilde{a} = \begin{cases} (n \cdot a^{(1)}, n \cdot a^{(2)}, n \cdot a^{(3)}), & n \geq 0 \\ (n \cdot a^{(3)}, n \cdot a^{(2)}, n \cdot a^{(1)}), & n < 0 \end{cases} \quad (3.6)$$

- Invers, definisinya sebagai berikut (Kwang H. Lee, 2005):

$$-\tilde{a} = (-a^{(3)}, -a^{(2)}, -a^{(1)}) \quad (3.7)$$

Karena perbedaan bentuk data tersebut, operasi-operasi pada tahapan selanjutnya juga akan mengalami berbagai perubahan. Perubahan-perubahan yang harus dilakukan yaitu:

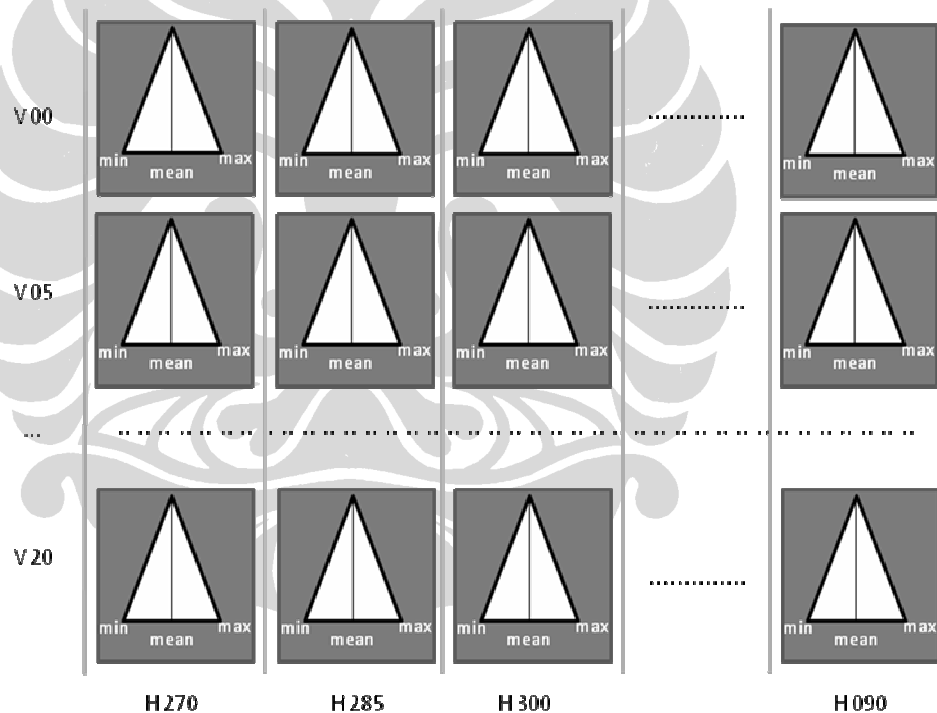
1. Pada tahap ekstraksi data, di sini awal terbentuknya data *fuzzy*. Perlakuan terhadap data hampir sama dengan algoritma yang memakai data rata-rata, hanya saja di algoritma ini tidak hanya informasi rata-rata yang diambil, tetapi juga informasi nilai minimal dan maksimal dari masing-masing dimensi pada masing-masing sudut. Pada tahap ini tidak dapat diberlakukan PCA, paling tidak sejauh ini.
2. Pada tahap pembuatan data acuan baru dengan menggunakan interpolasi. Persamaan interpolasi linier dan Bezier kuadratik hanya dapat dipakai untuk data berbentuk *crisp* (satuan). Maka pada tahap ini persamaan interpolasi

dengan mengganti operasi aritmetika bilangan biasa diubah menjadi operasi aritmetika bilangan *fuzzy* segitiga.

3. Pada tahap penentuan sudut tebakan, jarak Euclid tidak dapat diterapkan pada data berbentuk *fuzzy* segitiga, diusulkanlah suatu cara untuk menentukan jarak antara dua vektor *fuzzy* segitiga.

3.2.3 Pembentukan Data *Fuzzy* Segitiga

Sebenarnya tahap-tahap pembentukan data hampir sama dengan algoritma yang menggunakan data rata-rata per sudut. Perbedaannya hanya pada algoritma ini nilai minimal dan maksimal masing-masing dimensi pada masing-masing sudut juga diambil. Skema ekstraksi data pada sistem ini menjadi seperti Gambar 3.13.



Gambar 3.13 Skema Ekstraksi Data *Fuzzy* berdasarkan Sudut Horizontal

Gambar 3.13 menggambarkan pengelompokan berdasarkan sudut horizontal. Pada gambar tersebut terlihat bahwa pada setiap dimensi dalam setiap sudut terdapat 1 angka *fuzzy* yang diwakili 3 nilai *crisp* (minimal, rata-rata, dan maksimal).

Pada data uji, data masuk satu demi satu, sehingga untuk membuat data uji yang berbentuk *crisp* ini menjadi berbentuk *fuzzy*, data minimal diambil dari data uji dikurangi 5 dan data maksimal diambil dari data uji ditambah 5, sementara data rata-rata (tengah) adalah data uji itu sendiri.

3.2.4 Pembuatan Data Acuan *Fuzzy* Baru Menggunakan Interpolasi *Fuzzy*.

Metode interpolasi *fuzzy* yang dimaksud di penelitian ini adalah sebatas mengganti operasi-operasi aritmetika *crisp* (angka satuan biasa) menjadi operasi-operasi aritmetika *fuzzy*. Operasi aritmetika *fuzzy* itu sendiri sudah dipaparkan di Subbab 3.2.2.

Untuk persamaan interpolasi linier dengan data *crisp* dapat dilihat pada persamaan 2.1. Sedangkan persamaan interpolasi linier yang dimodifikasi untuk data *fuzzy* dapat ditulis seperti Persamaan 3.8.

$$\vec{P}_t = \vec{P}_0 + (\vec{P}_1 - \vec{P}_0)t \quad (3.8)$$

Pada Persamaan 3.8, semua operand yang merupakan titik adalah bilangan *fuzzy* segitiga. Oleh karena itu operasi penjumlahan, pengurangan, dan perkalian dengan skalar yang ada di dalamnya menggunakan operasi-operasi aritmetika *fuzzy* segitiga yang sudah dibahas pada subbab sebelumnya.

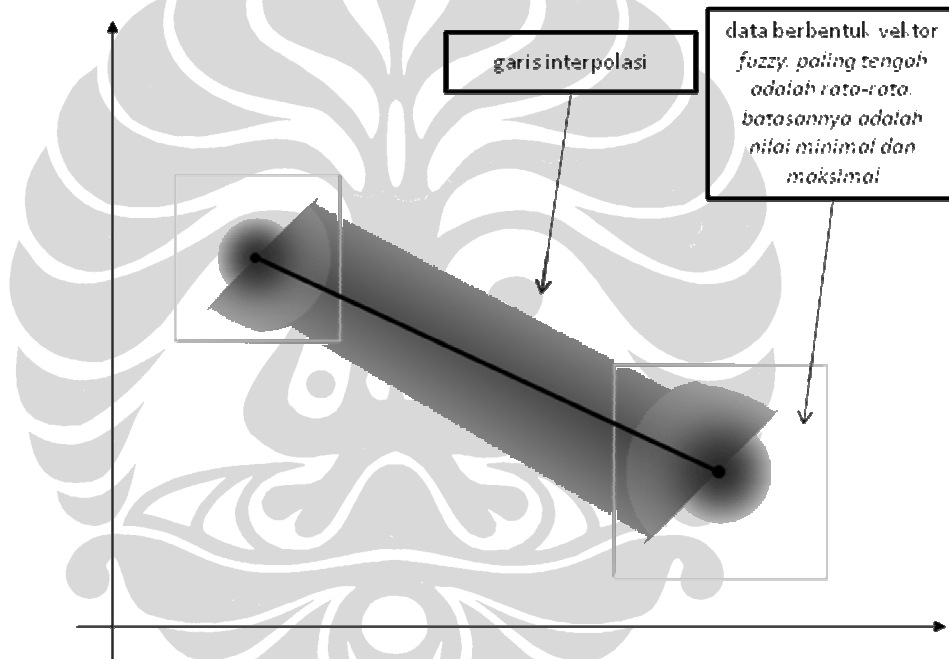
Sementara itu persamaan interpolasi Bezier kuadrat dengan data *crisp* dapat dilihat pada Persamaan 2.2. Sedangkan persamaan interpolasi Bezier kuadrat yang dimodifikasi untuk data *fuzzy* dapat ditulis seperti persamaan 3.9.

$$\vec{P}_t = (1-t)^2 \vec{P}_0 + 2t(1-t) \vec{P}_1 + t^2 \vec{P}_2 \quad (3.9)$$

Sedangkan untuk penempatan titik kontrol kedua dapat ditulis seperti Persamaan 3.10.

$$\tilde{P}_1 = \frac{\tilde{P}_t - ((1-t)^2\tilde{P}_0 + t^2\tilde{P}_2)}{2t(1-t)} \quad (3.10)$$

Pada kedua persamaan 3.10, sama dengan persamaan interpolasi linier, semua operand yang merupakan titik dianggap adalah bilangan *fuzzy* segitiga. Oleh karena itu operasi penjumlahan, pengurangan, dan perkalian dengan skalar yang ada di dalamnya menggunakan operasi-operasi aritmetika *fuzzy* segitiga yang sudah dibahas pada subbab sebelumnya. Gambar 3.14 menggambarkan interpolasi yang menggunakan data berbentuk *fuzzy*.



Gambar 3.14 Penggambaran Interpolasi dengan Data *Fuzzy*

3.2.5 Penentuan Tebakan Sistem serta Penghitungan Tingkat Ketepatan Tebakan

Sebenarnya algoritma penentuan tebakan tidak berbeda dengan algoritma yang menggunakan data rata-rata, yaitu data uji dihitung jaraknya dengan data acuan yang berkaitan, lalu dicari yang terdekat, itulah tebakan dari implementasi algoritma. Bagian yang berbeda adalah perhitungan jarak antara data uji dengan data acuan.

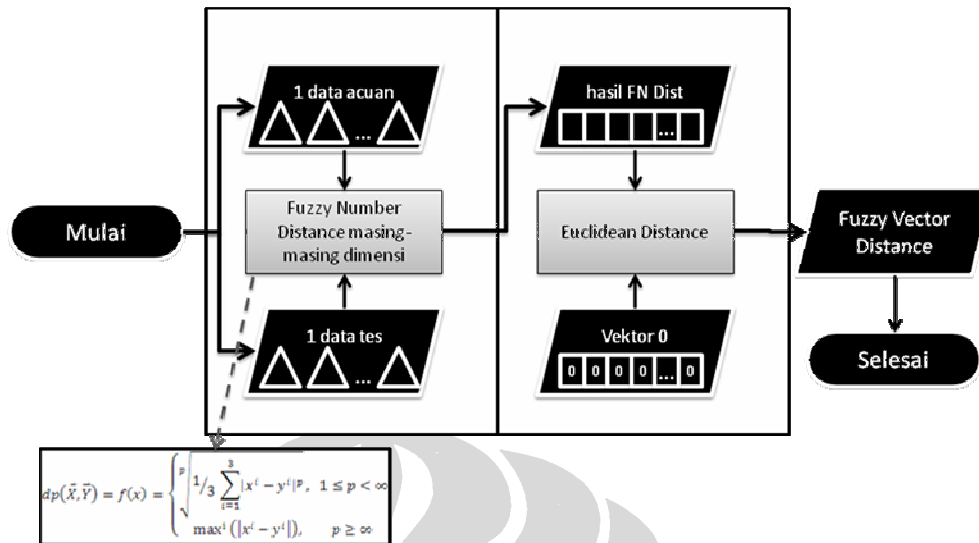
Pada algoritma dengan data rata-rata, bentuk datanya hanya merupakan vektor angka *crisp* biasa, sehingga penghitungan jaraknya dapat langsung menggunakan jarak Euclid. Sedangkan pada algoritma dengan data *fuzzy* segitiga, jarak yang dimaksud adalah jarak antara dua vektor *fuzzy*. Sejauh ini cara yang sudah dipublikasikan untuk menghitung jarak antara dua vektor *fuzzy* masih belum ditemukan. Yang ditemukan hanya persamaan untuk menghitung jarak antara dua angka *fuzzy* segitiga, yaitu seperti yang dituliskan pada persamaan 3.11.

$$d_p(\vec{X}, \vec{Y}) = \begin{cases} \sqrt[p]{\frac{1}{3} \sum_{i=1}^3 |x^{(i)} - y^{(i)}|^p}, & 1 \leq p < \infty \\ \max^{(i)} (|x^{(i)} - y^{(i)}|), & p \geq \infty \end{cases} \quad 3.11$$

Persamaan 3.11 merupakan modifikasi persamaan jarak angka *fuzzy* trapesium yang ditulis di *paper* Heilpern (1995). Variabel p pada persamaan di atas merupakan variabel bebas.

Pada dua metode penentuan jarak antar dua vektor *fuzzy* yang berbasis pada persamaan jarak antar dua angka *fuzzy* (persamaan 3.11) dikembangkan.

Metode pertama disebut Metode Jarak Vektor *Fuzzy* 1. Metode ini tidak memerlukan modifikasi dari persamaan 3.11. Namun persamaan tersebut akan menjadi bagian dari dua tahapan sampai akhirnya menghasilkan sebuah angka *crisp* yang merupakan jarak. Gambar 3.15 merupakan tahap-tahap penghitungan jarak antara dua vektor *fuzzy*.



Gambar 3.15 Skema Metode Jarak Vektor *Fuzzy* 1

Pada Gambar 3.15 terlihat bahwa metode ini membutuhkan dua tahap:

1. Menghitung jarak angka *fuzzy* (*Fuzzy Number Distance*) masing-masing dimensi. Hasil dari tahap ini adalah sebuah vektor *crisp* yang berisi masing-masing dimensi jarak angka *fuzzy*.
2. Hasil pada tahap pertama dihitung jarak Euclid-nya dengan vektor yang berisi angka nol di semua dimensi. Hal ini dilakukan dengan asumsi bahwa semakin dekat jarak kedua data pada masing masing dimensi, semakin dekat hasil penghitungan angka *fuzzy* dengan angka nol. Hasil akhir dari metode ini adalah sebuah angka *crisp* yang merupakan jarak antara dua vektor *fuzzy*.

Metode pertama disebut Metode Jarak Vektor *Fuzzy* 2. Metode ini merupakan metode sederhana yang memodifikasi persamaan 3.11. Pengurangan angka yang ada di dalamnya diasumsikan sebagai jarak Euclid antara dua vektor satu dimensi. Lalu vektor *fuzzy* dianggap sebagai tiga vektor yang terpisah, yaitu vektor minimal, vektor rata-rata, dan vektor maksimal sedemikian hingga persamaan yang dimodifikasi menjadi seperti persamaan 3.13. *komen:sip, maksud gue gini*

$$d_p(\vec{x}, \vec{y}) = \begin{cases} \sqrt[p]{\frac{1}{3} \sum_{i=1}^3 Ed(\vec{x}, \vec{y})^p}, & 1 \leq p < \infty \\ \max^{(i)} (Ed(\vec{x}, \vec{y})), & p \geq \infty \end{cases} \quad 3.13$$

Variabel Ed pada persamaan 3.13 merupakan persamaan jarak Euclid seperti pada persamaan 2.9. Sementara variabel p merupakan variabel bebas, pada eksperimen yang dilakukan, dipakailah $p = 1$ dan $p = 2$. Keluaran dari persamaan 3.13 adalah sebuah angka *crisp* yang merupakan jarak dari dua vektor *fuzzy*.

3.3 Pemberian Distorsi (*Noise*) pada Gambar

Selama ini masukan untuk implementasi algoritma, baik gambar untuk data acuan ataupun untuk data uji selalu merupakan gambar yang cukup ideal dan bersih. Sementara pada keadaan sebenarnya, data gambar yang bersih atau tanpa distorsi, tidak selalu didapatkan, sehingga jika implementasi algoritma hanya diberi masukan gambar yang bersih pada akhirnya tidak dapat mewakili keadaan sebenarnya.

Oleh karena itu dalam penelitian yang dilakukan, dijalankan juga eksperimen dengan data gambar terdistorsi sebagai data acuan dan atau sebagai data uji, baik dengan bentuk data rata-rata maupun data *fuzzy* segitiga. Sementara itu distorsi yang dikenai pada gambar merupakan empat jenis distorsi yang paling umum ditemui, yaitu Gauss, Poisson, *Salt and Pepper*, dan *Speckle*.

3.3.1 Gauss

Distorsi Gauss (*Gaussian Noise*) merupakan bentuk *white noise* dengan distribusi yang normal. *White noise* itu sendiri merupakan fluktuasi acak dari sinyal (dalam hal ini *grayscale*). Untuk mengenai distorsi jenis ini ke dalam gambar, cukup dengan menambahkan matriks gambar dengan matriks yang berisi angka acak yang terdistribusi normal. Dalam eksperimen, rata-rata dan variansi standar dari fungsi `imnoise(image, 'Gauss')` MATLAB digunakan, yaitu nilai rata-rata 0 dan variansi 0,01 (McAndrew, 2004).

3.3.2 Poisson

Distorsi ini diberikan kepada gambar sesuai dengan distribusi Poisson.

3.3.3 Salt and Pepper

Distorsi jenis ini dapat dianggap sebagai gangguan acak yang ekstrem pada data, yaitu titik-titik hitam atau putih yang tidak berhubungan dengan keadaan sinyal di sekitarnya. Dalam eksperimen, intensitas standar yang digunakan adalah intensitas dari fungsi `imnoise(image, 'salt & pepper')` MATLAB, yaitu nilai intensitas 0,1 atau 10 persen dari keseluruhan data (Denceux dan Masson, 2004).

3.3.4 Speckle

Distorsi *speckle* sedikit banyak memiliki persamaan dengan distorsi *Gauss*, yaitu berupa angka acak yang terdistribusi normal. Perbedaan dengan distorsi *Gauss* adalah untuk mengenai data gambar, angka-angka acak tersebut harus dikalikan dengan gambar. Dalam eksperimen, rata-rata dan variansi standar dari fungsi `imnoise(image, 'speckle')` MATLAB digunakan, yaitu nilai rata-rata 0 dan variansi 0,04 (Denceux dan Masson, 2004).

Gambar 3.16 menunjukkan foto-foto yang dikenai distorsi.



Gambar 3.16 Foto-foto yang Dikenai Distorsi

Skema pemberian distorsi akan dibahas secara rinci pada Bab 4. Tahap ekstraksi data pada algoritma, dengan data rata-rata ataupun *fuzzy* segitiga, merupakan tahap dimana gambar diberi distorsi, tepatnya sebelum informasi *grayscale* dari gambar dibaca oleh implementasi algoritma. Setelah itu tidak ada perbedaan dengan algoritma sebelumnya.

