

Bab 6

Probabilistic Neural Network (PNN) Multinomial Dengan Algoritme Expectation Maximization (EM)

Metodologi klasifikasi citra yang dikembangkan menggunakan *Probabilistic Neural Network (PNN) model multinomial* yang dioptimalkan dengan algoritme *Expectation Maximization (EM)*. Selanjutnya pengklasifikasi uniform optimal ditempatkan pada sekma fusi data dan deteksi perubahan wilayah untuk menyelesaikan masalah pengolahan citra.

6.1 Metodologi Klasifikasi Uniform PNN Model Multinomial

Banyak metode klasifikasi citra berdasarkan tekstur yang didahului dengan menggunakan dasar pengukuran statistik pada *Gray Level Co-occurrence Matrik (GLCM)* [6]. Berdasarkan beberapa studi terdahulu, dapat dinyatakan bahwa teknik co-occurrence matrik memberikan fitur yang baik untuk klasifikasi citra terurama untuk citra sensor optik dan *Synthetic Aperture Radar (SAR)*. Dalam penelitian ini akan digunakan co-occurrence matrik sebagai masukan pengklasifikasi citra dengan metode NN yang identik dengan PNN. Berdasarkan fungsi diskriminan, Master (1995) menyatakan bahwa fungsi kerapatan probabilitas pada PNN dapat didekati oleh fungsi kerapatan apapun yang tidak hanya terbatas pada Gaussian [22]. Dalam penelitian ini akan digunakan fungsi kerapatan probabilitas *model multinomial* yang bersifat lebih adaptif dan akomodatif terhadap karakteristik data yang kompleks [14].

Probabilistic Neural Network (PNN) Gaussian yang dikembangkan Donald dkk. tahun 1990 berhasil dengan baik untuk data citra sensor optik, sedangkan untuk data citra sensor SAR memberikan hasil yang kurang memuaskan [27].

Berdasarkan hal di atas, penelitian disertasi ini menggunakan fungsi kerapatan *multinomial* berdasarkan model *Gray Level Co-occurrence Matrix* (GLCM), dan selanjutnya struktur *neural network* yang dikembangkan disebut PNN *model multinomial*. GLCM merupakan suatu model yang sering digunakan dalam pekerjaan analisis tekstur. Model ini merupakan transformasi suatu citra tekstur menjadi citra dengan karakteristik yang bersifat homogen. Hal ini dilakukan diantaranya agar mudah dalam pekerjaan klasifikasi atau interpretasi citra. Matrik *co-occurrence* berisi informasi tentang seberapa sering atau frekuensi kemunculan suatu piksel i dengan nilai *gray level* g_i berada berdampingan atau bersamaan dengan piksel j dengan nilai *gray level* g_j secara horizontal atau Barat-Timur (0°), vertikal atau Utara-Selatan (90°), diagonal atau Timur Laut – Barat Daya atau Barat Laut – Tenggara (45° atau 135°), dan berada pada jarak $d = 1, 2, \dots$. Berikut ini contoh perhitungan matrik *co-occurrence* [6]:

Misal terdapat sejumlah piksel yang memiliki *gray level* pada rentang 0 – 5 dengan ukuran 5 x 5 sebagai berikut:

3	3	4	4	7
5	4	5	3	3
2	3	3	2	3
5	4	6	2	3
4	3	3	4	4

Pada arah 0° dan jarak $d = 1$ (berdampingan langsung) ditentukan melalui langkah-langkah sebagai berikut:

- 1). Tentukan banyaknya *gray level* yang berbeda dalam citra tersebut, kemudian urutkan dari kecil ke besar. Untuk citra di atas, terdapat 6 *gray level* yang berbeda, yaitu 2, 3, 4, 4, 6, dan 7.

2). Bentuklah matrik C berukuran k x k dengan k adalah banyaknya *gray level*, sedangkan elemen a_{ij} -nya menyatakan jumlah kemunculan piksel dengan *gray level* g_i muncul berdampingan dengan piksel dengan *gray level* g_j pada arah 0° dimana $1 \leq i, j \leq k$. Untuk citra di atas $k = 6$ dan $g_1=2, g_2=3, g_3=4, g_4=5, g_5=6,$ dan $g_6=7$. Maka matrik C adalah sebagai berikut:

$$C = \begin{bmatrix} 0 & 3 & 0 & 0 & 0 & 0 \\ 1 & 4 & 2 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 & 1 & 1 \\ 0 & 1 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3). Probabilitas occurrence (normalisasi) diperoleh:

$$p(u) = \begin{bmatrix} 0.02500 & 0.01371 & 0.05000 & 0.01000 & 0.01000 & 0.01000 \\ 0.05000 & 0.00482 & 0.05000 & 0.01000 & 0.01000 & 0.01000 \\ 0.02500 & 0.05000 & 0.05000 & 0.05000 & 0.05000 & 0.05000 \\ 0.02500 & 0.05000 & 0.05000 & 0.01000 & 0.01000 & 0.01000 \\ 0.05000 & 0.04074 & 0.05000 & 0.01000 & 0.01000 & 0.01000 \\ 0.02500 & 0.04074 & 0.05000 & 0.01000 & 0.01000 & 0.01000 \end{bmatrix}$$

Selanjutnya sebuah citra dapat direpresentasikan sebagai matrik dari probabilitas p_{ij} berdasarkan co-occurrence $u=(u_{1,1}, u_{1,2}, \dots, u_{m,m})$ yang dimodelkan dengan fungsi kerapatan *multinomial* sebagai berikut [21]:

$$p(u) = n! \prod_{i,j} \frac{p_{ij}^{u_{ij}}}{u_{ij}!} \dots\dots\dots (6.1)$$

$$\sum_{i,j} p_{ij} = 1 \text{ dan } \sum_{i,j} u_{ij} = n \dots\dots\dots (6.2)$$

Keterangan :

$p(u)$: probabilitas occurrence kelas u

i, j : nomor gray level

n : jumlah kemunculan piksel pada kelas u

$p_{ij}^{u_{ij}}$: probabilitas occurrence piksel dengan gray level i berdampingan dengan gray level j dari kelas u

u_{ij} : frekuensi kemunculan piksel pada kelas u dengan gray level i berdampingan dengan gray level j

Berikut ini permumusan model distribusi *multinomial*, misal kemungkinan kejadian kelas $C_1, C_2, C_3, \dots, C_J$ sebagai berikut [26]:

kejadian C_1 : n_1 kali

kejadian C_2 : n_2 kali

.

.

.

kejadian C_J : n_J kali.

Didefinisikan banyaknya occurrence untuk setiap kelas = $\{n_1, n_2, n_3, \dots, n_J\}$.

Probabilitas occurrence dinyatakan:

$$p_1, p_2, p_3, \dots, p_J : P(n_1, n_2, n_3, \dots, n_J; p_1, p_2, p_3, \dots, p_J) \dots\dots\dots (6.3)$$

Selanjutnya probabilitas occurrence dinyatakan dengan P_o dan N_o .

$$P_o = p_1^{n_1} \cdot p_2^{n_2} \cdot p_3^{n_3} \dots p_J^{n_J} = \prod_{j=1}^J p_j^{n_j} \dots\dots\dots (6.4)$$

$$N_o = \frac{N!}{n_1!(N-n_1)!} \cdot \frac{(N-n_1)!}{n_2!(N-n_1-n_2)!} \dots \frac{(N-n_1-\dots-n_{j-1})!}{n_j!(N-n_1-n_2-\dots-n_j)!} \dots (6.5)$$

$$N_o = \frac{N!}{n_1!n_2!\dots n_J!} \dots\dots\dots (6.5)$$

Probabilitas *occurrence* dinyatakan sebagai berikut :

$$P(n_1, n_2, \dots, n_J; p_1, p_2, \dots, p_J) = P_o N_o = \frac{N!}{n_1!n_2!\dots n_J!} p_1^{n_1} \cdot p_2^{n_2} \dots p_J^{n_J} \dots\dots\dots (6.6)$$

$$P(n_1, n_2, \dots, n_J; p_1, p_2, \dots, p_J) = P_o N_o = \frac{N!}{\prod_{j=1}^J n_j!} \prod_{j=1}^J p_j^{n_j} \dots\dots\dots (6.7)$$

$$P(n_1, n_2, \dots, n_J; p_1, p_2, \dots, p_J) = P_o N_o = N! \prod_{j=1}^J \frac{p_j^{n_j}}{n_j!} \dots\dots\dots (6.8)$$

Misal X adalah sebuah fitur, $p(u)$ adalah probabilitas *prior* kelas u , dan $p(u|x)$ adalah probabilitas *posterior* kelas u dari fitur x , maka hubungan fitur x dengan kelas u dapat dinyatakan sebagai berikut [21]:

$$C(x, u) = \log \frac{p(u/x)}{p(u)} \dots\dots\dots (6.9)$$

Berdasarkan persamaan dibawah ini :

$$p(x) = \frac{N_x}{N} \dots\dots\dots (6.10)$$

$$p(ux) = \frac{N_{ux}}{N} \dots\dots\dots (6.11)$$

$$p(u | x) = \frac{N_{ux}}{N_x} \dots\dots\dots (6.12)$$

$$p(ux) = \frac{N_{ux}}{N} \frac{N_x}{N_x} \dots\dots\dots (6.13)$$

$$p(ux) = \frac{N_{ux}}{N_x} \frac{N_x}{N} \dots\dots\dots (6.14)$$

$$p(ux) = p(u | x)p(x) \dots\dots\dots (5,15)$$

$$p(u | x) = \frac{p(ux)}{p(x)} \dots\dots\dots (6.16)$$

$$p(ux) = p(u | x)p(x) = p(x | u)p(u) \dots\dots\dots (6.17)$$

$$p(u | x) = \frac{p(ux)}{p(x)} = \frac{p(x | u)p(u)}{p(x)} \dots\dots\dots (6.18)$$

$$\frac{p(u/x)}{p(u)} = \frac{p(x/u)}{p(x)} \dots\dots\dots (6.19)$$

Maka

$$C(x,u) = \log \frac{p(u/x)}{p(u)} = \log \frac{p(x/u)}{p(x)} \dots\dots\dots (6.20)$$

Jika p_{ij} adalah probabilitas *occurrence* dari citra sumber, p_{ij}^u adalah probabilitas *occurrence* dari kelas obyek u, dan x_{ij} menyatakan probabilitas *occurrence* data uji maka :

$$\frac{p(x|u)}{p(x)} = \prod_{i,j} \left(\frac{p_{ij}^u}{p_{ij}} \right)^{x_{ij}} \dots\dots\dots (6.21)$$

atau

$$\log \frac{p(x|u)}{p(x)} = \sum_{i,j} x_{ij} (\log p_{ij}^u - \log p_{ij}) \dots\dots\dots (6.22)$$

Aturan keputusan fitur X masuk ke dalam kelas u jika :

$$\sum_{i,j} x_{ij} (\log p_{ij}^u - \log p_{ij}) > \sum_{i,j} x_{ij} (\log p_{ij}^v - \log p_{ij}) \quad \dots\dots\dots (6.23)$$

Keterangan :

- i,j : nomor gray level
- P_{ij}^u : Probabilitas occurrence kelas pelatihan u
- P_{ij}^v : Probabilitas occurrence kelas pelatihan v
- P_{ij} : Probabilitas occurrence sumber
- x_{ij} : Probabilitas occurrence kelas uji

6.2 Arsitektur dan Algoritme PNN Model Multinomial

Terdiri atas 4 unit lapisan: unit masukan, unit pola, unit jumlah, dan unit keputusan.

- Unit masukan terhubung ke semua unit pola.
- Unit pola terhubung ke unit jumlah dari kelas yang sama.
- Unit jumlah terhubung ke unit kaputusan.
- Unit keputusan memutuskan kelas (label) dari masukan.

Algoritme Pelatihan :

Langkah-1: Rekuantisasi citra utuh ke dalam sejumlah gray level (6).

Langkah-2: Hitung deskriptor co-occurrence citra utuh (P_{ij}).

Langkah-3: Untuk setiap kelas pelatihan, lakukan langkah 4 -9

Langkah-4: Tentukan koordinat dan ukuran kelas pelatihan.

Langkah-5: Hitung deskriptor co-occurrence kelas pelatihan (P_{ij}^u)

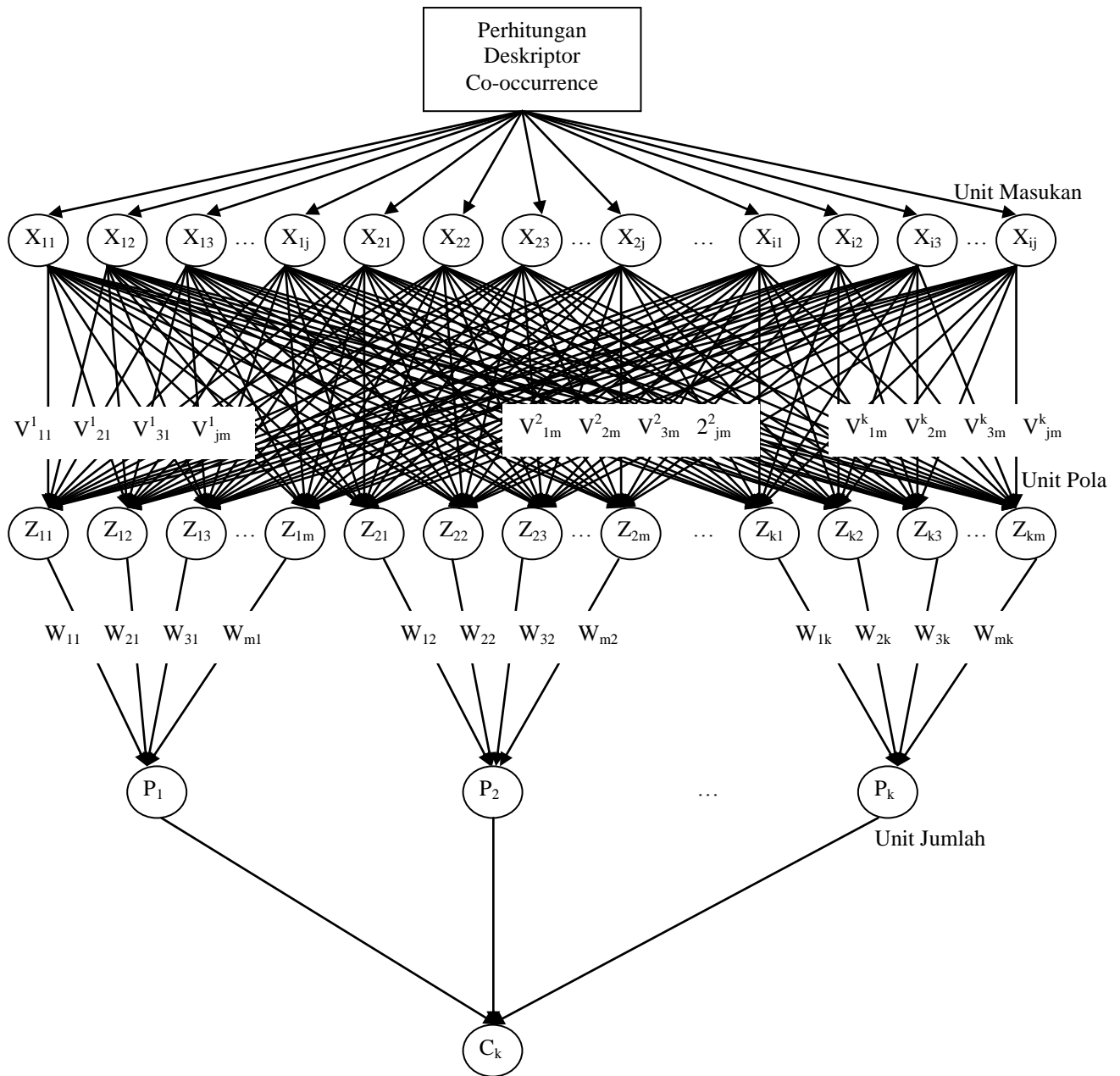
Langkah-6: Buat unit masukan (X_{ij}).

Langkah-7: Buat unit pola sebanyak sampel pelatihan (m) setiap kelas (k).

Langkah-9: Lakukan pembagian co-occurrence kelas pelatihan oleh co-occurrence citra dan nyatakan sebagai bobot unit masukan dengan unit pola (V_{jm}^k).

Langkah-10: Buat unit jumlah sebanyak k kelas sampel pelatihan dan hubungkan dengan unit pola dengan bobot koneksi 1 ($w_{mk}=1$).

Langkah-11: Buat unit keputusan dan hubungkan dengan unit jumlah.



X_{ij} : elemen ke- ij dari co-occurrence masukan X
 V^k_{jm} : bobot neuron masukan ke j untuk neuron pola m dari kelas k
 W_{mk} : bobot neuron pola ke m untuk neuron jumlah kelas k

Gambar-6.1 : Arsitektur PNN model Multinomial

Algoritme Pengujian:

Langkah-1: Untuk setiap piksel citra uji, lakukan langkah-2 – langkah-4.

Langkah-2: Tentukan ukuran occurrence.

Langkah-3: Hitung deskriptor co-occurrence piksel uji (x_{ij}) sebagai sinyal masukan.

Langkah-4: Untuk setiap unit pola m pada setiap kelas, jumlahkan hasil kali sinyal masukan dengan bobot yang bersesuaian:

$$net_{km} = \sum_{i,j} x_{ij} (\log p_{ij}^u - \log p_{ij}).$$

kemudian hitung aktivasinya dengan fungsi aktivasi:

$$z_{km} = f(net_{km})$$

dan meneruskan sebagai sinyal masukan unit jumlah.

Langkah-5: Setiap unit jumlah k , jumlahkan hasil kali sinyal masukan dengan bobotnya:

$$net_k = \sum_{m=1} z_{km} \cdot w_{mk}$$

kemudian hitung aktivasinya dengan fungsi aktivasi:

$$P_k = f(net_k)$$

dan meneruskan sebagai sinyal masukan unit keputusan.

Langkah-6: Unit keputusan melakukan pemilihan nilai maksimum dari unit jumlah :

$$C(x) = \arg \max\{P_k(x)\},$$

C : kategori obyek dari piksel uji x

kemudian melakukan klasifikasi.

6.3 Optimalisasi Probabilitas *Posterior*

Probabilitas *posterior* yang dihasilkan pengklasifikasi perlu dioptimalkan dengan algoritme *Expectation Maximization (EM)* agar merupakan representasi yang optimal dari suatu data input. Keputusan klasifikasi *Probabilistic Neural Network (PNN) model multinomial* menggunakan nilai maksimum probabilitas *posterior* oleh *neuron keputusan* berdasarkan *neuron somasi*, dan peranan EM adalah untuk mengoptimalkan probabilitas *posterior* sebelum proses keputusan, maka proses optimalisasi berada diantara *unit pola* dan *unit somasi*.

Proses *algoritme EM* melalui tahap ekspektasi dan maksimisasi. Tahap ekspektasi dapat dinyatakan dengan persamaan [17] :

$$f^c(\omega_i | x) = \frac{p^c(\omega_i)p^c(x|\omega_i)}{\sum_{i=1}^L p(\omega_i)p(x|\omega_i)} \dots\dots\dots (6.24)$$

Keterangan :

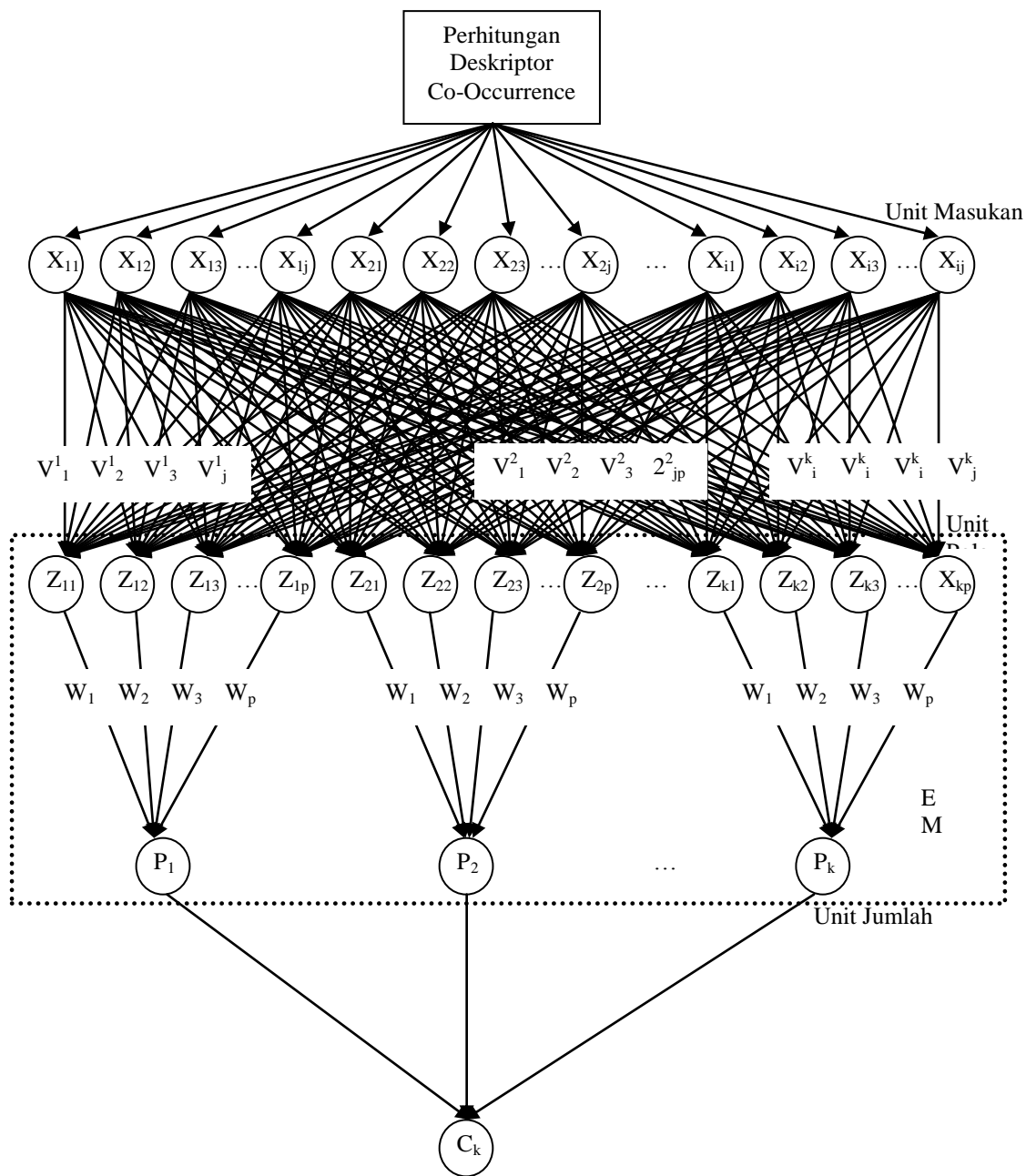
- $f(\omega_i | x)$: probabilitas *posterior* masukan x untuk kelas i
- $p(\omega_i)$: probabilitas *prior* kelas i
- $p(x|\omega_i)$: pdf masukan x untuk kelas i
- i : nomor kelas
- c : *current*

Tahap maksimalisasi dinyatakan dengan persamaan :

$$p^+(\omega_i | x) = \arg \max_{\omega \in \Omega} \sum_{i=1}^L p(\omega_i) \ln p(x|\omega_i) \dots\dots\dots (6.25)$$

Keterangan :

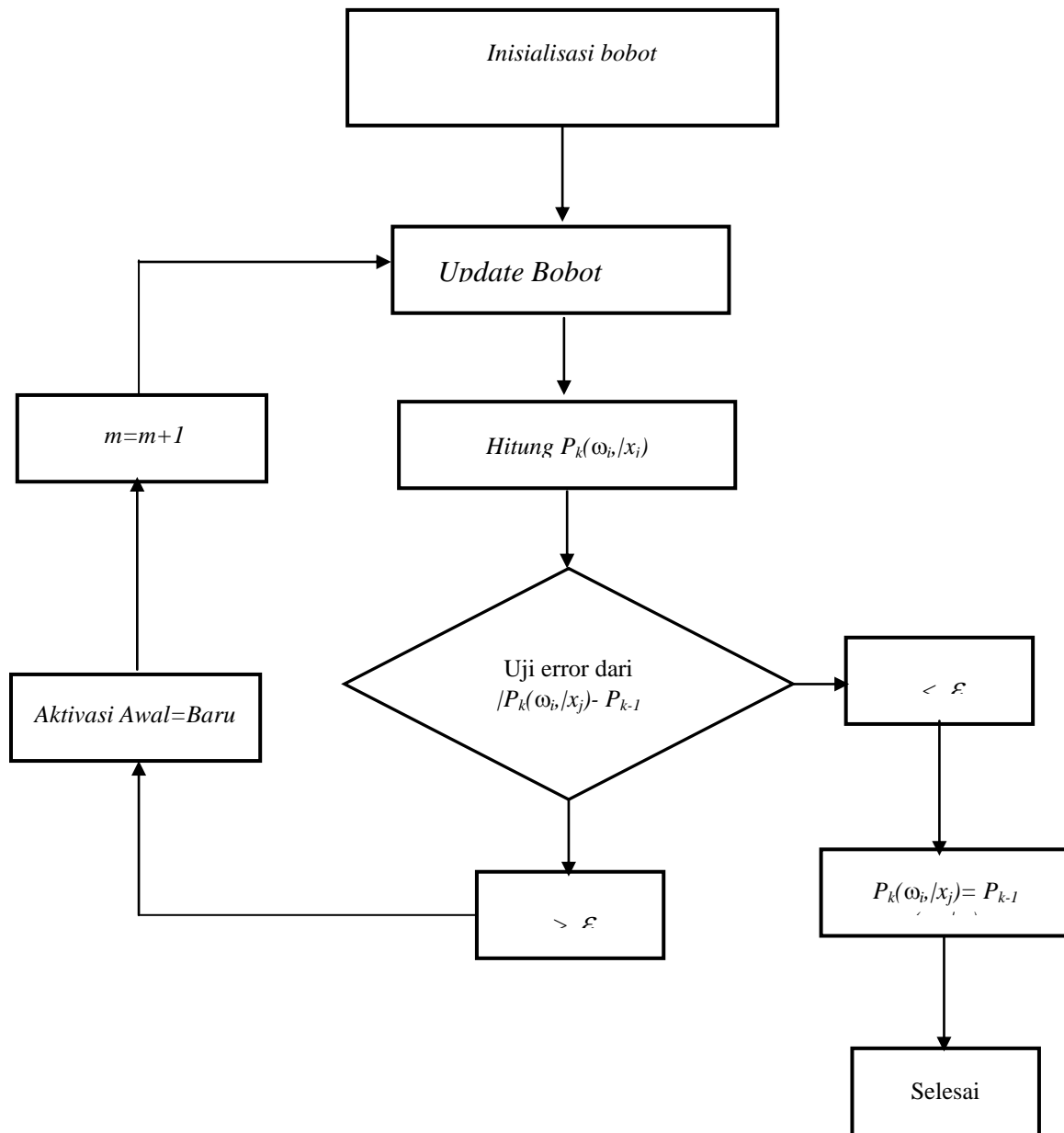
- i : nomor kelas
- $+$: estimasi selanjutnya (*next*)



X_{ij} : elemen ke- ij dari co-occurrence masukan X
 V_{jp}^k : bobot neuron masukan ke j untuk neuron pola p dari kelas k
 W_{pk} : bobot neuron pola ke p untuk neuron jumlah kelas k

Gambar-6.2 : Arsitektur PNN model Multinomial dan EM

Algoritme EM pada pengklasifikasi PNN *model multinomial* memerlukan proses tambahan (pembaharuan) merupakan adaptasi bobot unit *pola* dengan unit somasi



Gambar-6.3 : Diagram Alir Estimasi Probabilitas *Posterior*

yang semula diinisialiasi dengan nilai 1 (yang memiliki interpretasi bahwa informasi dari *unit pola* hanya diteruskan langsung). Dengan adanya proses EM, bobot koneksi ini mengalami penyesuaian sehingga keluaran *neuron somasi* yang dihasilkan merupakan representasi yang optimal.

6.4 Optimalisasi Probabilitas *Joint Prior*

Probabilitas *Joint Prior* (JP) merupakan probabilitas gabungan $P(\omega_i, v_j)$ piksel kelas i dan kelas j dari dua sumber yang diperlukan untuk mengestimasi sebuah kelas obyek. Perumusan gabungan dua sumber dapat dinyatakan sebagai berikut [5]:

$$p(\omega_i, v_j | X_1, X_2) = \frac{p(\omega_i | X_1) p(v_j | X_2)}{p(\omega_i) p(v_j)} p(\omega_i, v_j) \dots\dots\dots (6.26)$$

Keterangan :

- $p(\omega_i | X_1)$: probabilitas posterior pola X untuk kelas i dari sumber-1
- $p(v_j | X_2)$: probabilitas posterior pola X untuk kelas j dari sumber-2
- $p(\omega_i)$: probabilitas prior kelas i dari sumber-1
- $p(v_j)$: probabilitas prior kelas j dari sumber-2
- $p(\omega_i, v_j)$: probabilitas joint prior kelas i dan j dari sumber-1 dan 2
- i, j : nomor kelas obyek

Probabilitas $P(w_i, v_j)$ didefinisikan sebagai elemen-elemen dari matrik JP (ukuran $M_1 \times M_2$), yang dihitung dengan maksimalisasi persamaan berikut:

$$L(JP) = \prod_{q=1}^S \left(\sum_{\omega n \in \Omega} \sum_{vm \in \mu} P(\omega n, vm) P(X_1^q, X_2^q / \omega n, vm) \right) \dots\dots\dots (6.27)$$

Keterangan :

- q : nomor piksel yang di klasifikasi
- X_k^q : piksel ke q dari citra I_k
- n, m : nomor kelas

Persamaan rekursi untuk mengestimasi $P(w_i, v_j)$ dapat dinyatakan sebagai berikut :

$$P_{k+1}(\omega_i, v_j) = \frac{1}{S} \sum_{q=1}^S \frac{P_k(\omega_i, v_j) P(X_1^q, X_2^q / \omega_i, v_j)}{\sum_{\omega n \in \Omega} \sum_{vm \in \mu} P(\omega n, vm) P(X_1^q, X_2^q / \omega n, vm)} \dots(6.28)$$

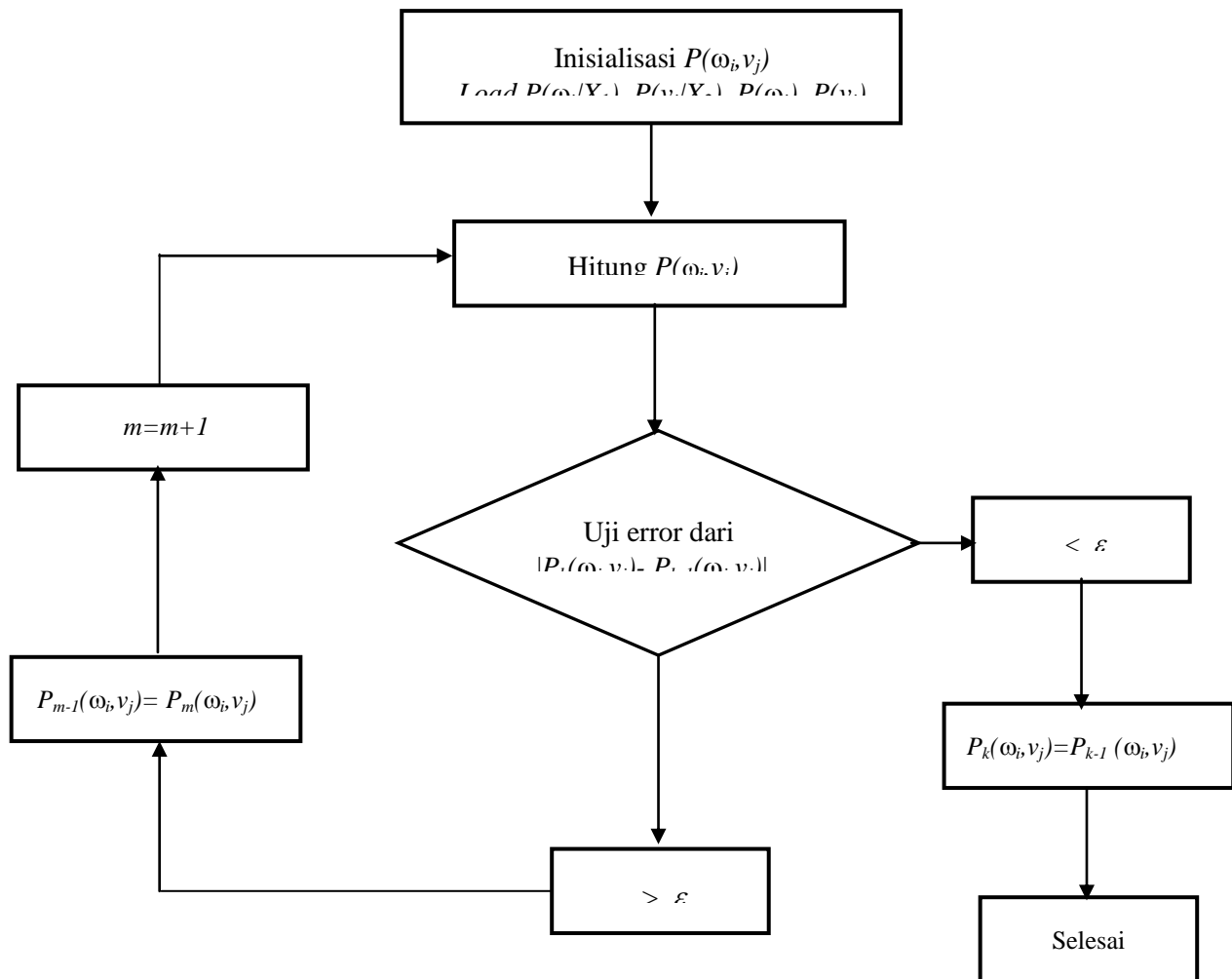
atau dapat ditulis sebagai berikut :

$$P_{k+1}(\omega_i, v_j) = A_{ij} \sum_{q=1}^S \frac{P(\omega_i, v_j) P(\omega_i / X_1^q) P(v_j / X_2^q)}{\sum_{\omega n \in \Omega} \sum_{vm \in \mu} \frac{P_k(\omega n, vm)}{P_k(\omega n) P_k(vm)} P(\omega_i / X_1^q) P(v_j / X_2^q)} \dots(6.29)$$

dan
$$A_{ij} = \frac{1}{SP(\omega_i)P(v_j)} \dots\dots\dots(6.30)$$

Keterangan:

- q : nomor piksel yang di klasifikasi
- X_k^q : piksel ke q dari citra I_k .
- $P_k(\omega_i, v_j)$: probabilitas joint prior kelas i dan j pada iterasi ke- k .
- $p(\omega_i | X_1)$: probabilitas posterior kelas i dari sumber-1
- $p(v_j | X_2)$: probabilitas posterior kelas j dari sumber-2
- $p(\omega_i)$: probabilitas prior kelas i dari sumber-1
- $p(v_j)$: probabilitas prior kelas j dari sumber-2
- $p(\omega_i, v_j)$: probabilitas joint prior kelas i dan j dari sumber-1 dan 2
- i, j, m, n : nomor kelas obyek



Gambar-6.4 : Diagram Alir Estimasi Probabilitas Joint Prior

Pada tahap awal, elemen matrik diinisialisasi dengan memberikan probabilitas yang sama pada setiap pasangan kelas dengan persamaan :

$$P_0 = \frac{1}{M_1 M_2} \quad \forall \omega_i \in \Omega, v_j \in \mu \dots \dots \dots (6.31)$$

Algoritme EM di-iterasi sampai konvergen yaitu dicapainya selisih $P(\omega_i, v_j)$ diantara estimasi pada dua iterasi berturut-turut maksimum dibawah suatu ambang. Maka kriteria henti interasi didefinisikan sebagai berikut :

$$\max_{\omega_i, v_j} |p_{k+1}(\omega_i, v_j) - p_k(\omega_i, v_j)| < \varepsilon, \quad \omega_i \in \Omega, \quad v_j \in \mu \dots \dots \dots (6.32)$$

$$\varepsilon \in [0,1]$$

Selanjutnya probabilitas *joint prior* yang diperoleh pada iterasi akhir digunakan sebagai parameter untuk deteksi perubahan wilayah.