

BAB 2

DASAR TEORI

2.1 KAJIAN PUSTAKA

Penelitian [7] membahas tentang pengklasifikasian pergerakan tangan dan kaki berbasis sinyal EEG. Sinyal EEG digunakan sebagai pengendali pergerakan devais bionik berbasis BCI. Metode yang digunakan untuk rancang sistem yaitu *Common Spatial Patterns* (CSP) untuk tahap ekstraksi fitur dan *stochastic gradient descent* dalam *Multilayer Perceptron backpropagation* (MLP-BP) untuk mengklasifikasi tangan dan kaki. Pengujian sistem menggunakan *confusion Matrix* kepada sepuluh subjek dan berbagai jumlah node. Rata-rata tertinggi akurasi rancangan sistem mencapai 94,38% pada sistem dengan delapan *node* terhadap *hidden layer*. Perbedaan dengan penelitian penulis yaitu pada metode yang digunakan untuk mengklasifikasi sinyal EEG seperti *Convolutional Neural Network* (CNN).

Penelitian [4] membahas tentang pengklasifikasian sinyal EEG pada pergerakan jari manusia. Ada dua bagian rancangan sistem klasifikasi diantaranya yaitu *convolutional layer* dan *multilayer perceptron* yang nantinya diimplementasikan menggunakan Python 3.7 dengan *library* TensorFlow 2.0 (Keras). Rancang sistem diuji dengan lima subjek dari data MI-EEG 5F menggunakan frekuensi pencuplikan 200 Hz. Pengujian ini melibatkan *Kfold-cross validation* dan analisis terhadap *confusion matrix*. Dari hasil pengujian ini, akurasi tertinggi didapatkan oleh rancangan dengan jumlah kernel 50 sebesar 51,711%.

Penelitian [5] membahas tentang klasifikasi sinyal EEG *Motor Imagery* tangan kanan dan kiri dengan fitur *Common Spatial Pattern* menggunakan *Multilayer Perceptron Backpropagation*. Pada penelitian ini menggunakan dataset sekunder dari *BCI Competition IV (2b)* dengan 9 subjek penelitian. Pengujian dilakukan dengan menggunakan 1 dataset dengan berbagai jumlah node lapisan tersembunyi seperti 8, 16, dan 24 pada setiap saluran gelombang. Berdasarkan pengujian, nilai akurasi merupakan rata-rata akurasi tertinggi dari 10 K-Fold eksperimen. Akurasi pada 8 node lapisan tersembunyi adalah 68,5%, 16 node lapisan tersembunyi adalah 68,5%, dan 24 node lapisan tersembunyi adalah 68,7% sekaligus menjadi hasil terbaik pada 3

skenario tersebut. Perbedaan dengan penelitian penulis yaitu pada metode yang digunakan untuk mengklasifikasi sinyal EEG seperti *Convolutional Neural Network* (CNN).

Penelitian [8] menggunakan Elektromiografi (EMG) sebagai teknik penelitian yang bersangkutan dengan perekaman sinyal *myoelectric*. Penelitian ini bertujuan untuk mengidentifikasi pergerakan jari tangan manusia melewati sinyal EMG menggunakan cara klasifikasi. Jenis gelombang yang diklasifikasi dalam penelitian ini ada 4, diantaranya yaitu *Literal*, *Grasp*, *Fist/Hook* dan *Tip*. Penelitian ini menggunakan metode *Deep Neural Network – Stacked Denoising Autoencoder*. Hasil pengujian ini mendapatkan nilai akurasi tertinggi sebesar 94%. Penelitian melakukan proses klasifikasi menggunakan sinyal Elektromiografi (EMG), berbeda dengan penelitian penulis yang menggunakan sinyal Elektroensefalogram (EEG).

Penelitian [9] membahas tentang pendeteksian pergerakan jari menggunakan fitur *Power Spectral Density* (PSD) Burg. Perekamn sinyal EEG dilakukan dengan frekuensi *sampling* 1000 Hz. Sinyal dianalisis dengan cara dibagi menjadi tiga segmen yaitu 1000 ms, 500 ms dan 250 ms. Ekstraksi fitur dan pengenalan nola dilakukan dengan menggunakan *Common Average Reference* (CAR) dan *Support Vector Machine* (SVM). Hasil dari penelitian ini menunjukkan bahwa system dapat melakukan deteksi pergerakan jari dengan nilai akurasi $\pm 65,37\%$ dengan Panjang sampel 1000 ms. Rangkuman tinjauan Pustaka dapat dilihat pada Tabel 2.1

Tabel 2.1 Perbandingan Penelitian Yang Menjadi Tinjauan Pustaka

Nama Peneliti	Metode	Hasil	Perbedaan Metode
Rahmat Widadi, Dodi Zulherman,	<i>Common Spatial Patterns</i> (CSP) untuk tahap ekstraksi fitur dan <i>Stochastic Gradient Descent</i> dalam <i>Multilayer Percepton backpropagation</i> (MLP-BP) untuk	Dengan menggunakan 3 metode pada sistem klasifikasi EEG, mampu mendapatkan nilai akurasi rata-rata tertinggi mencapai 94,38% pada sistem dengan delapan node pada <i>hidden layer</i> .	Metode yang digunakan penulis yaitu <i>Convolutional Neural Network</i> Dua Dimensi dengan menggunakan tiga kali proses konvolusi

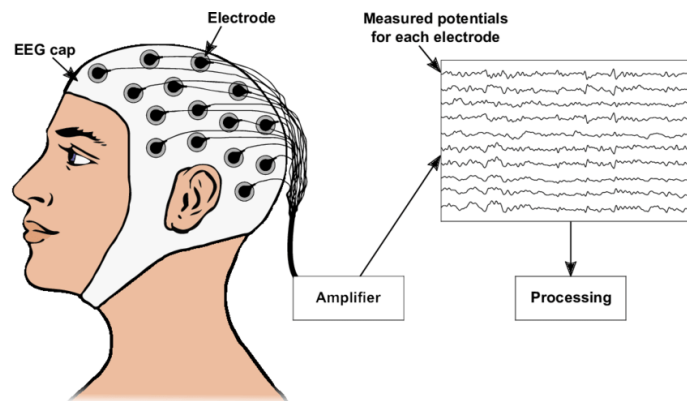
Nama Peneliti	Metode	Hasil	Perbedaan Metode
	mengklasifikasi tangan dan kaki, serta <i>Confusion Matrix</i> sebagai pengujian system [7].		
Rahmat Widadi, Bongga Arif Widodo, Dodi Zulherman	<i>Convolutional Neural Network</i> satu dimensi untuk klasifikasi sinyal EEG [4].	Dengan menggunakan data MI-EEG 5F dan frekuensi pencuplikan 200 Hz, serta melibatkan <i>Kfold-cross validation</i> sebagai pengujian dan <i>confusion matrix</i> sebagai analisis, mendapatkan akurasi tertinggi pada kernel 50 sebesar 51,711%.	Metode yang digunakan penulis yaitu <i>Convolutional Neural Network</i> Dua Dimensi dengan menggunakan tiga kali proses konvolusi
Synatria Subekti	<i>Common Spatial Patterns</i> (CSP) untuk tahap ekstraksi fitur dan <i>Multilayer Perceptron backpropagation</i> (MLP-BP) untuk mengklasifikasi tangan kanan dan kiri [5].	Dengan menggunakan dataset sekunder dari BCI Competition IV (2b) dengan 9 subjek penelitian, mampu mendapatkan nilai akurasi tertinggi di 24 node lapisan tersembunyi sebesar 68,7%.	Metode yang digunakan penulis yaitu <i>Convolutional Neural Network</i> Dua Dimensi dengan menggunakan tiga kali proses konvolusi
Echa Pangersa Sugianto Oeoen, Jondri, Untari Novia Wisety	<i>Deep Neural Network - Stacked Denoising Autoencode</i> untuk mengklasifikasi 4 jenis gelombang [8].	Dengan menggunakan metode <i>Deep Neural Network - Stacked Denoising Autoencoder</i> , mendapatkan nilai akurasi tertinggi sebesar 94%.	Metode yang digunakan penulis yaitu <i>Convolutional Neural Network</i> Dua Dimensi
Reza Darmakusuma, Ary S. Prihatmanto, Adi Indriyanto,	<i>Power Spectral Density</i> untuk mendeteksi pergerakan jari serta <i>Common Average Reference</i>	Hasil pada penelitian ini menunjukkan bahwa sistem dapat mendeteksi pergerakan jari tangan dengan akurasi \pm 65,37%	Metode yang digunakan penulis yaitu <i>Convolutional Neural Network</i> Dua Dimensi

Nama Peneliti	Metode	Hasil	Perbedaan Metode
Tati L. mengko	dan <i>SupportVector Machine</i> untuk mengekstrak fitur dan pengenalan pola [9].	dengan panjang sampel 1000 ms.	dengan menggunakan tiga kali proses konvolusi

2.2 DASAR TEORI

2.2.1 Electroencephalograph (EEG)

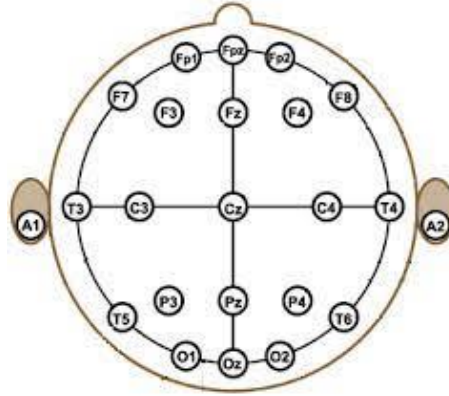
Electroencephalograph (EEG) merupakan proses perekaman aktivitas listrik pada otak manusia. Pada perkembangannya, mekanisme perekaman dilakukan dengan cara menempelkan sejumlah elektroda pada bagian atas kepala (*scalps*). Berdasarkan aktivitas listrik pada otak, penempatan kanal dibagi menjadi: *frontal* (F) untuk pengontrolan, pengenalan, gerakan, dan kemampuan bicara. *Parietal* (P), untuk menerima informasi temperature, posisi tubuh, dan rangsangan sentuhan. *Occipital* (O) untuk menerima rangsangan visual dan tulisan. *Temporal* (T) untuk menerima informasi rangsangan telinga yang berhubungan dengan memori [10].



Gambar 2.1 *Electroencephalograph* [11]

Tinggi nya resolusi sinyal pada EEG bisa didapatkan dengan memodifikasi jumlah elektroda, kontak elektroda, sensor, filter dan system nirkabel. Maka dari itu terdapat sistem pada penempatan elektroda EEG. Sistem penempatan elektroda pada bagian kulit kepala sudah ditetapkan secara internasional oleh *International Federation of Societes of Electroencephalogram* yang dikenal dengan istilah “*International Electrode Placement System*” atau bisa disebut sebagai sistem 10-20. Sistem penempatan pada elektroda 10-20 mengatur tentang letak titik penempatan

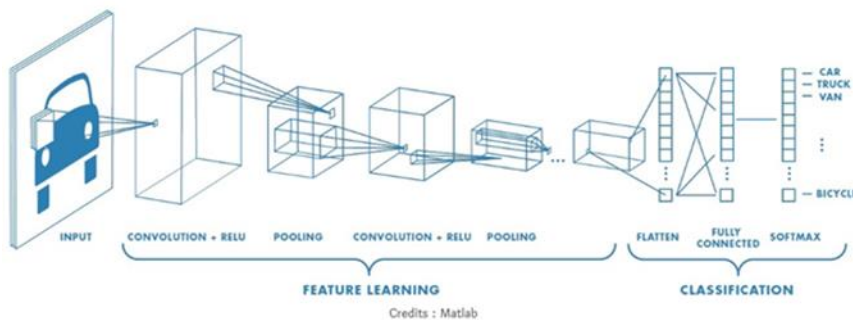
elektroda pada bagian kulit kepala. Perbandingan jarak 10% digunakan pada elektroda pertama dan terakhir serta interval 20% untuk elektroda lainnya yang berada disepanjang garis utama [12].



Gambar 2.2 Sistem Penempatan Elektroda Pada Kulit Kepala [13]

2.2.2 Convolutional Neural Network

Convolutional neural network merupakan sebuah metode dari *machine learning* yang dikembangkan dari *Multi-Layer Perceptron* (MLP) yang dirancang untuk mengolah serta membuat data dari dua dimensi. CNN juga merupakan salah satu metode yang memiliki tingkat jaringan serta mempunyai banyak penerapan yang ada didalam citra. CNN memiliki metode diantaranya yaitu klasifikasi menggunakan *feedward* dan tahap pembelajarannya yang menggunakan *backpropagation*. Pada metode ini juga memiliki prinsip kerja seperti mempunyai kesamaan dengan metode MLP, setiap neuronnya yang tetap ada didalam metode CNN dan disajikan dalam bentuk dua dimensi yang dimana tidak sama dengan metode MLP yang setiap neuronnya hanya memiliki ukuran satu dimensi [14].



Gambar 2.3 Arsitektur *Convolutional Neural Network* [15]

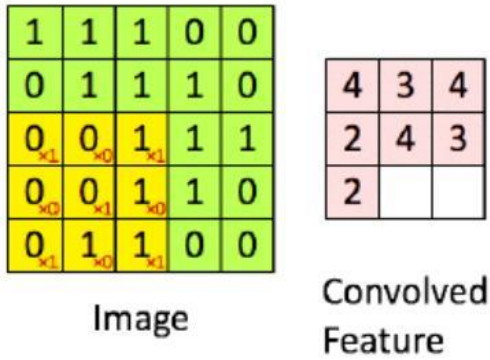
Seperti yang dilihat pada Gambar 2.3 CNN menerima inputan berupa gambar yang nantinya akan diolah inputanya melalui *convolution layer*, *RELU* dan *pooling layer* agar dapat diklasifikasi, selanjutnya data dari inputan akan diklasifikasi menggunakan *flatten*, *fully connected layer* dan *softmax*. Ada dua bagian besar yang ada pada arsitektur CNN yaitu *Feature Extraction Layer* dan *Classification Layer*. Pada bagian *Feature Extraction layer* terjadi “*encoding*” dari image menjadi *feature map* berupa angka-angka yang menampilkan citra tersebut (*Feature Extraction*). *Feature extraction layer* terdiri dari dua bagian yaitu *convolution layer* dan *pooling layer*. Bagian pertama pada metode CNN yaitu *convolution layer* yang menjadi pembeda dengan *neural network* yang lain [16].

2.2.3 Feature Extraction Layer

Terdapat dua lapisan pada *feature extraction layer* yang berguna sebagai penerima masukkan (*input*) gambar secara langsung dan diolah sampai menghasilkan keluaran (*output*) data multi dimension *array*. Terdapat lapisan yang ada pada proses ini diantaranya lapisan konvolusi dan lapisan *pooling*, dimana pada setiap proses lapisan akan menghasilkan *feature maps* dalam bentuk angka yang menampilkan gambar dan kemudian akan diteruskan ke bagian lapisan klasifikasi [16].

2.2.3.1 Convolution layer

Convolution layer merupakan lapisan yang berguna untuk menjalankan operasi konvolusi terhadap output layer sebelumnya. Pada *Convolutional Neural Network* (CNN), *layer* ini merupakan blok utama yang didalamnya berisis *filter-filter* yang dapat dipelajari secara acak agar dapat melakukan operasi konvolusi dengan tujuan sebagai ekstraksi fitur agar representasi fitur dari input layer dapat dipelajari. Melakukan operasi konvolusi pada data citra bertujuan untuk mengekstraksi fitur dari input citra, operasi konvolusi tersebut dapat menghasilkan transformasi linear pada data yang diinput sesuai informasi spasial yang tersedia pada data. Pada *layer* tersebut bobot akan menspesifikasikan *kernel* konvolusi yang akan digunakan, agar *kernel* konvolusi bisa dilatih berdasarkan *input* pada *Convolutional Neural Network* (CNN) [17].



Gambar 2.4 Contoh Operasi Matematis Proses Konvolusi [18]

Gambar 2.4 merupakan contoh operasi konvolusi antara matriks citra masukan dengan matriks filter. Nantinya filter akan digeser keseluruhan permukaan citra sehingga dapat menghasilkan keluaran matriks yang disebut *feature map*. *Feature map* didapatkan dari rumus berikut.

$$n_{out} = \left(\frac{n_{out} - K + 2p}{s} \right) + 1 \quad (2.1)$$

Dimana:

n_{out} : Ukuran *feature map*

n_{in} : Ukuran matriks masukan

k : Ukuran matriks filter

p : Ukuran *padding*

s : *Stride*

2.2.3.2 ReLU Layer

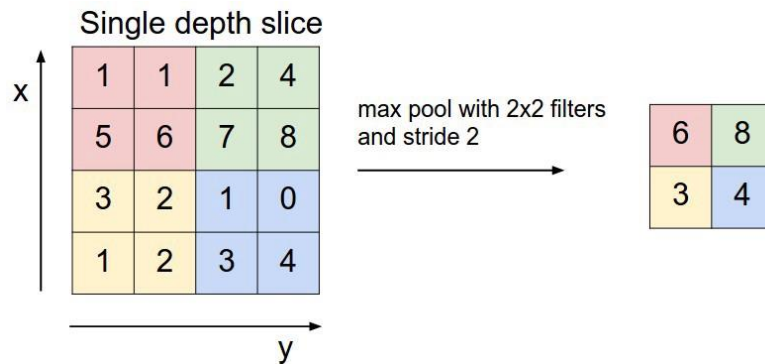
ReLU (*Rectified Linear Unit*) merupakan fungsi aktivasi yang biasa digunakan dikarenakan sifatnya yang berfungsi dengan baik. Fungsi aktivasi ReLU secara umum dinyatakan dalam persamaan berikut :

$$f(x) = \max(0, x) \quad (2.2)$$

Berdasarkan dari persamaan di atas, nilai *output* dari neuron dapat dinyatakan sebagai 0 apabila *input* nya bernilai negatif. Apabila nilai *input* dari fungsi aktivasi bernilai positif maka *output* dari *neuron* merupakan nilai input itu sendiri [16].

2.2.3.3 Pooling layer

Merupakan lapisan yang berfungsi sebagai pengurangan ukuran spasial pada fitur konvolusi agar berkurangnya sumber daya komputasi yang dibutuhkan agar bisa memproses data melalui pengurangan dimensi pada *feature map* (*downsampling*) sehingga dapat mempercepat komputasi karena semakin sedikit nya parameter yang diperbarui. *Pooling layer* juga berguna untuk mengekstrak fitur dominan agar proses pelatihan model lebih efektif. *Pooling layer* memiliki dua jenis yaitu *max pooling* dan *average pooling*. Fungsi dari *max pooling* yaitu mengembalikan nilai maksimum pada bagian gambar yang dicakup oleh kernel, sedangkan fungsi dari *average pooling* yaitu mengembalikan nilai rata-rata pada bagian gambar yang dicakup oleh kernel [17]. *Pooling* yang digunakan yaitu *max* dan *average pooling*. Jika menggunakan *max pooling* 2x2 dengan stride 2, maka pada setiap pergeseran filter, nilai maksimum pada area 2x2 pixel tersebut yang akan dipilih, sedangkan yang memilih rata-rata nya yaitu *average pooling* [16].



Gambar 2.5 Contoh Operasi Matematis Proses *Max Pooling* [16]

Gambar 2.5 merupakan contoh operasi matematis dari proses *max pooling* yang dimana proses nya mirip dengan *convolution layer* hanya pada *max pooling* biasanya tidak menggunakan *padding*.

$$n_{out} = \left(\frac{n_{out-K}}{s} \right) + 1 \quad (2.3)$$

Dimana:

n_{out} : Ukuran *feature map*

n_{in} : Ukuran matriks masukan

k : Ukuran matriks filter

s : *Stride*

2.2.4 *Classification Layer*

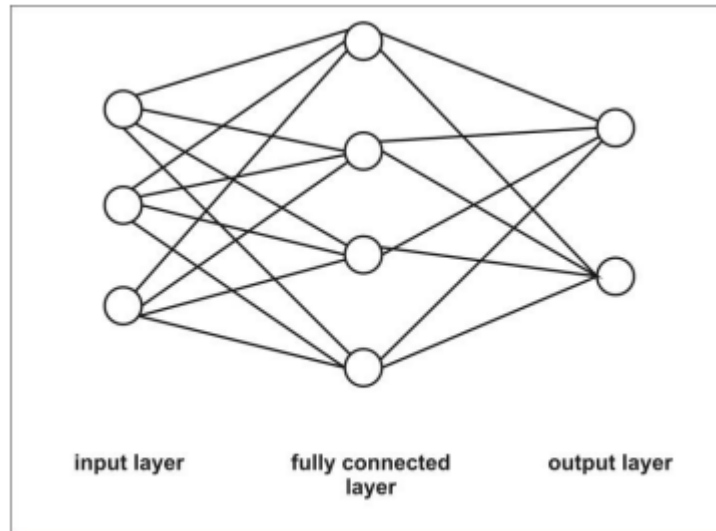
Lapisan ini terdiri dari beberapa lapisan yang memiliki neuron yang yang terhubung seluruhnya (*fully connected*) dengan lapisan lain. Lapisan ini akan menerima masukan (*input*) dari output *output layer* pada bagian *feature learning* yang selanjutnya akan di proses oleh *flatten* dengan cara menambahkan beberapa *header layer* pada *fully connected* yang akan menghasilkan keluaran berupa nilai akurasi dari setiap kelas [16].

2.2.4.1 *Fully connected layer*

Input dari hasil *output pooling layer* diambil oleh *fully connected layer* yang berupa *future map*. *Future map* ini masih dalam bentuk multidimensional array, maka dari itu lapisan ini harus melakukan *reshape future map* dahulu dan akan menghasilkan vektor sebanyak n -dimensi yang dimana n merupakan jumlah kelas output yang akan dipilih program [19]. Secara umum *fully connected layer* merupakan proses akhir dari arsitektur yang dimana prosesnya ditampilkan sebagai perkalian dari matriks sederhana yang dilanjutkan dengan menambahkan vector bias serta penerapan fungsi non-linear sebagai berikut:

$$y = f(W^T x + b) \quad (2.4)$$

Dimana y sebagai nilai vektor pada *output* yang memiliki fungsi dari W berupa matriks sederhana dengan bobot koneksi antar unit yang selanjutnya dikalikan dengan x sebagai sebuah vektor *input* yang kemudian ditambah dengan b sebagai vector bias [16].



Gambar 2.6 *Fully Connected Layer* [19]


2.2.4.2 Flatten

Flatten merupakan proses pembentukan ulang fitur (*reshape feature map*) menjadi sebuah vektor agar dapat digunakan sebagai input dari *fully connected layer* [20].

2.2.5 K-Fold Cross Validation

k-fold cross Validation merupakan salah satu teknik untuk mengestimasi tingkat kesalahan dengan cara kerja *k-fold validation* yaitu mengelompokkan data latih dan data uji, selanjutnya melakukan proses pengujian sebanyak k kali. Contoh (pada gambar 2.10) dalam 10-fold *cross validation*, data dibagi menjadi 10 *fold* berukuran sama, sehingga ada 10 subset data untuk mengevaluasi kinerja algoritma. Dari 10 subset data, 9 *fold* akan digunakan sebagai pelatihan dan 1 *fold* sebagai pengujian [21].

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10



Gambar 2.7 10 Fold Cross Validation [21]

Pada Gambar 2.7 seluruh data dibagi secara acak menjadi K buah subset B_k dengan ukuran yang sama, B_k merupakan himpunan dari bagian $\{1, \dots, n\}$ sehingga $\cup_{k=1}^K B_k = \{1, \dots, n\}$ dan $B_j \cap B_k = \emptyset (j \neq k)$ kemudian dilakukan literasi sebanyak K kali. Pada literasi ke k, subset B_k menjadi *test set*, sedangkan subset yang lain menjadi *training set*. Selanjutnya rata-rata akurasi akan dihitung menggunakan K buah literasi dengan persamaan sebagai berikut [21].

$$akursi = \frac{\sum \text{data uji benar klasifikasi}}{\sum \text{total data uji}} \times 100 \quad (2.5)$$

2.2.6 Multi-Class Confusion Matrix

Confusion matrix merupakan matriks dua dimensi yang dimana barisnya menunjukkan label sesungguhnya dan menunjukkan kolom label yang diprediksi oleh pengklasifikasi. Membuat *confusion matrix* untuk multi-class sederhana dan lugas seperti yang di bawah ini.

$$M(r, c) = \sum_{i=1}^m (I(y_i = r)I(h(x_i) = c)), \forall r, c \in \{0, \dots, q - 1\} \quad (2.2)$$

Dimana M merupakan matriks konfusi, sedangkan r dan c merupakan baris dan kolom pada matriks konfusi, m merupakan jumlah instance dalam kumpulan data uji, $I(\cdot)$ merupakan fungsi indikator, x_i merupakan input i^{th} terhadap pengklasifikasian $h(\cdot)$, y_i merupakan label sebenarnya yang diberikan kepada input x_i , dan q merupakan jumlah kelas. Algoritma 1 menunjukkan metode untuk membuat *confusion matrix* untuk pengklasifikasian *multi-class*.

Algorithm 1 Multi-Class Confusion Matrix**For** each input instance **do** r = assigned label c = predicted label $M(r,c) + = 1$ **end for**

Multi-class confusion matrix menampilkan distribusi prediksi pada semua kelas dalam satu tampilan ringkas. Dengan menormalkan baris *confusion matrix*, presentase *FN* pada setiap kelas yang sama dengan setiap baris yang diperoleh. Sementara matriks ternormalisasi bertujuan untuk mencari presentase prediksi benar atau salah, *confusion matrix* dengan jumlah perhitungan sebenarnya dari prediksi benar atau salah dapat memberikan informasi tentang ukuran setiap kelas dalam kasus data set yang seimbang [22].