BAB 2
TINJAUAN PUSTAKA

2.1 Ekstraksi Informasi

Ekstraksi informasi adalah suatu proses untuk mengubah informasi tidak terstruktur yang terdapat dalam teks ke dalam data terstruktur [6]. Data terstruktur dihasilkan dengan mengidentifikasi seperangkat konsep yang telah ditetapkan sebelumnya dalam *domain* tertentu, dengan mengabaikan informasi lain yang tidak relevan, dimana sebuah domain terdiri dari kumpulan teks bersamaan dengan kebutuhan informasi yang jelas [7]. Menurut Owda, Crockett, dan Lee, salah satu pendekatan yang dimiliki ekstraksi informasi adalah *automatic training*. Pada pendekatan *automatic training*, melakukan proses ekstraksi informasi tidak membutuhkan ahli untuk membuat aturan ekstraksi, melainkan membutuhkan seseorang yang mengetahui pada bidang tertentu untuk membuat mesin pembelajaran secara otomatis dari *text corpus* yang dibangun untuk mengekstraksi informasi [8].

Misalnya, salah satu contoh data seperti "PEMBANGUNAN APLIKASI PEMBELAJARAN INTERAKTIF MATA PELAJARAN TEKNOLOGI DASAR OTOMOTIF KELAS X DI SMK NEGERI 4 SUKABUMI" telah diberikan kategori sebagai Judul. Kemudian, dengan data tersebut mesin mempelajarinya sebagai kategori Judul. Ketika ada data baru yang mempunyai kemiripan seperti kategori Judul, mesin akan mengenalinya dan melakukan klasifikasi sebagai kategori Judul.

2.2 Dokumen Karya Tulis Ilmiah Skripsi

Dokumen adalah data, catatan, dan/atau keterangan [9]. Dokumen yang digunakan pada penelitian ini adalah dokumen karya tulis ilmiah skripsi Program Studi Teknik Informatika Universitas Komputer Indonesia. Dokumen dimana yang digunakan hanya lembar sampul dan abstrak.

Karya ilmiah adalah karya seorang ilmuwan (yang berupa hasil pengembangan) yang ingin mengembangkan ilmu pengetahuan, teknologi dan seni yang diperoleh dari kepustakaan, kumpulan pengalaman, dan pengetahuan orang

lain sebelumnya [10]. Karya ilmiah yang digunakan pada penelitian ini merupakan karya ilmiah mahasiswa lulusan skripsi Program Studi Teknik Informatika Universitas Komputer Indonesia.

Skripsi adalah karya ilmiah yang ditulis oleh mahasiswa sebagai bagian persyaratan pendidikan akademis di perguruan tinggi [11]. Pada dokumen skripsi digital biasanya tidak disertakan dokumen khusus untuk abstrak, sedangkan pada dokumen skripsi yang telah melalui proses dokumentasi, biasanya disertakan dengan dokumen abstrak.

Pada penelitian ini, dokumen karya tulis ilmiah skripsi digunakan sebagai *dataset* yang memiliki beberapa kategori pada lembar sampul dan abstrak. Jenis-jenis kategori didapat dari penelitian sebelumnya [1]. Pada lembar sampul terdiri dari kategori Judul Penelitian (Sampul), Jenis Penelitian, Kalimat Pengajuan, Penulis (Sampul), NIM (Sampul), Program Studi, Fakultas, Universitas, dan Tahun. Sedangkan pada lembar abstrak terdiri dari Judul Halaman Abstrak, Judul Penelitian (Abstrak), *Other*, Penulis (Abstrak), NIM (Abstrak), Isi Abstrak, dan Kata kunci. Sebagai gambaran jelasnya, berikut bagian – bagian kategori pada lembar sampul dan abstrak disertai dengan kelasnya dapat dilihat pada Tabel 2.1.

Tabel 2.1 Bagian – Bagian Kategori pada Lembar Sampul dan Abstrak Skripsi

Lembar sampul skripsi	No	Kategori	Kelas
	1	Judul Penelitian (Sampul)	0
	2	Jenis Penelitian	1
	3	Kalimat Pengajuan	2
	4	Penulis (Sampul)	3
	5	NIM (Sampul)	4
	6	Program Studi	5
	7	Fakultas	6
	8	Universitas	7

9

PEMBANGUNAN KAMUS JENIS KATA (1) **SEBAGAI SUMBER DAYA NLP BAHASA INDONESIA** **SKRIPSI** (2) Diajukan untuk Menempuh Ujian Akhir Sarjana (3) **CEPPY EFRAIM CHRISTIANTOSA G. BOLLY** (4) **10112904** (5)  **PROGRAM STUDI TEKNIK INFORMATIKA** (6) **FAKULTAS TEKNIK DAN ILMU KOMPUTER** (7) **UNIVERSITAS KOMPUTER INDONESIA** (8) **2016** (9)	9	Tahun	8

Lembar abstrak skripsi	No	Kategori	Kelas
ABSTRAK (9) **PEMBANGUNAN KAMUS JENIS KATA SEBAGAI SUMBER DAYA NLP BAHASA INDONESIA** (10) Oleh: (11) **CEPPY EFRAIM CHRISTIANTOSA G. BOLLY** (12) **10112904** (13) Kamus adalah buku referensi yang memuat daftar kata atau gabungan kata dengan keterangan mengenai pelbagai segi maknanya dan penggunaannya dalam bahasa, biasanya disusun menurut abjad. Selain dalam bentuk buku, saat ini kamus dijumpai berupa kamus digital yang dapat diakses secara online. Pengembangan kamus *online* Bahasa Indonesia tidak disertai dengan sumber daya, dengan kata lain pengguna hanya memakai dan tidak mempunyai hak akses untuk mengubah atau menambahkan. Sumber daya kamus berguna sebagai data masukan seperti kata dasar, jenis kata, *stopword,* korpus, dalam pengklasifikasian kelas kata pada proses *Question-Answering, language generator, information extraction, summarization, machine translation* dan lain-lain. Berdasarkan kebutuhan akan sumber daya maka, penelitian yang dilakukan adalah pembangunan kamus jenis kata yang diharapakan mampu menghasilkan sumber daya yang dapat digunakan dalam pengembangan bahasa dalam bidang NLP. (14) Sumber masukan yang dikelola adalah kbbi format .txt, dengan pengklasifikasian 7 kelas kata merunut Tata Bahasa Baku Bahasa Indonesia. Tahap pengambilan kata dan jenis kata terdiri dari sepuluh langkah yaitu penghilangan digit di awal kata, penghilangan spasi kosong di awal kalimat, penghilangan baris di awali simbol, penghilangan *blankline*, penghilangan spasi ganda, penghilangan simbol kecuali strip"-", penghilangan baris kurang dari dua kata, pengambilan kata dan jenis kata, pengecekan kata, dan pengurutan sehingga menghasilkan format kata diikuti jenis kata selanjutnya disimpan dalam *database*. Hasil akhir dari pembangunan kamus jenis kata adalah *database* kamus jenis kata dalam tiga format dan kamus online sebagai implementasi. Kamus yang dibangun tidak mengolah kata majemuk. Hasil pengujian pembangunan kamus jenis kata menghasilkan 38.870 lema. Selain dapat bebas menggunakan sumber daya kamus, diharapkan pengguna dapat mengembangkan sistem yang dibangun. Kata kunci : Kamus jenis kata, kamus online, pengklasifikasian kelas kata, *resource* NLP. (15)	1	Judul Halaman Abstrak	9
	2	Judul Penelitian (Abstrak)	10
	3	*Other*	11
	4	Penulis (Abstrak)	12
	5	NIM (Abstrak)	13
	6	Isi Abstrak	14
	7	Kata kunci	15

Setiap angka yang terdapat pada lembar sampul dan lembar abstrak menunjukkan bagian nomor kategori. Beberapa kategori diatas direpresentasikan menjadi kelas berbentuk angka untuk proses pembelajaran dan klasifikasi algoritma CNN. Total kelas yang akan digunakan pada penelitian ini sebanyak 16 kelas, dapat dikatakan pula dengan istilah *multi class*.

2.3 Ekstraksi Fitur

Menurut Prihatini, ekstraksi fitur merupakan proses untuk mencari nilai - nilai fitur yang terkandung dalam dokumen [12]. Fitur dapat diartikan sebagai ciri dari setiap data yang dikenali oleh sistem sehingga menghasilkan nilai fitur. Ekstraksi fitur merupakan topik penting dalam klasifikasi, karena fitur-fitur yang baik akan sanggup meningkatkan tingkat akurasi, sementara fitur-fitur yang tidak baik cenderung memperburuk tingkat akurasi [13].

Pada penelitian ini, ekstraksi fitur digunakan untuk memberikan bobot pada setiap token berdasarkan fitur yang telah ditentukan. Ekstraksi fitur yang digunakan sebanyak 15 fitur. 15 fitur merujuk pada penelitian yang dilakukan oleh Firdamdam [1]. Berikut penjelasan dari kelima belas fitur yang digunakan pada penelitian ini dapat dilihat pada Tabel 2.2.

Tabel 2.2 15 Ekstraksi Fitur

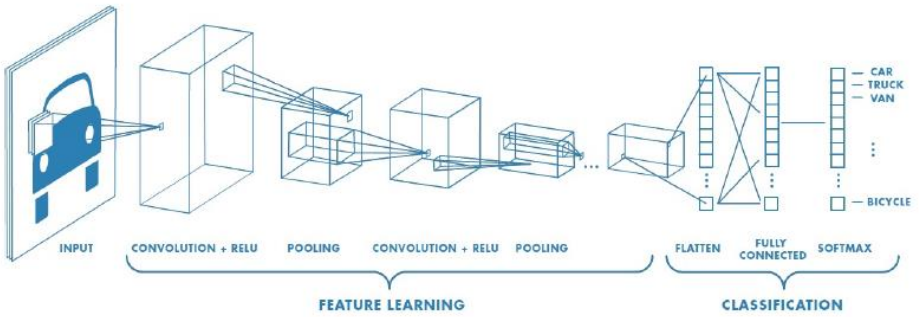
No	Ekstraksi Fitur	Keterangan
1	*INITCAPS*	Mengenali setiap token yang hurufnya diawali dengan kapital.
2	*ALLCAPS*	Mengenali setiap token yang semua hurufnya kapital.
3	*CONTAINSDIGIT*	Mengenali setiap token yang mengandung digit.
4	*ALLDIGIT*	Mengenali setiap token yang semuanya digit.
5	*CONTAINSDOTS*	Mengenali setiap token yang mengandung titik.
6	*LOWERCASE*	Mengenali setiap token yang semuanya huruf kecil.
7	*PUNCTUATION*	Mengenali setiap token yang mengandung tanda tertentu seperti titik, koma, titik dua, titik koma, tanda kurung, dan tanda seru.

No	Ekstraksi Fitur	Keterangan
8	*EIGHTDIGIT*	Fitur tambahan pada penelitian ini, fitur ini dikhususkan untuk mengenali token yang memiliki digit dengan panjang 8 digit.
9	*WORD*	Fitur tambahan pada penelitian ini, fitur ini dikhususkan untuk memberikan bobot pada token untuk kelas JENIS_PENELITIAN dan KALIMAT_PENGAJUAN.
10	*LINE_START*	Mengenali posisi token pada indeks array awal.
11	*LINE_IN*	Mengenali posisi token pada indeks array tengah.
12	*LINE_END*	Mengenali posisi token pada indeks array akhir.
13	*PERSON*	Mengenali token nama seseorang.
14	*ORGANIZATION*	Mengenali token sebuah organisasi.
15	*YEAR*	Mengenali ciri token tahun.

Setiap fitur pada Tabel 2.2 akan memberikan bobot 1 atau 0. Bobot 1 diberikan oleh setiap fitur jika token termasuk pada fitur. Begitupula dengan token yang tidak termasuk pada fitur, akan diberikan bobot 0.

2.4 *Convolutional Neural Network*

Deep learning adalah bagian dari *Artificial Neural Network* yang pertama kali diperkenalkan oleh Rina Dechter pada tahun 1986. *Convolutional Neural Network* (CNN) adalah salah satu bentuk *feed forward artificial neural network* yang sering digunakan untuk model dengan input berupa data gambar atau video [14]. *CNN* termasuk di dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. CNN terinspirasi oleh proses-proses biologi dimana pola konektivitas antar *neuron* menyerupai organisasi *visual cortex* pada binatang [15]. Arsitektur dari CNN terbagi menjadi 2 bagian besar yaitu *Feature Extraction Layer* dan *Fully-Connected Layer*. Seperti pada gambar 2.1 berikut:



Gambar 1.1 Arsitektur CNN

Semua lapisan pada *CNN* tersebut disusun secara bertumpuk-tumpuk, seperti sepotong *sandwich* yang terdiri atas roti bagian bawah, sayuran, daging, keju, saus tomat, mayonnaise, saus sambal dan roti bagian atas. Masukan pada *CNN* memiliki arsitektur 3 dimensi: lebar (*width*), tinggi (*height*), dan dalam (*depth*). Lebar dan tinggi dalam masukan citra menyatakan dimensi citra tersebut sedangkan dalamnya menyatakan kedalaman citra seperti kanal *Red*, *Green*, *Blue* pada citra *RGB*. Masukan tersebut kemudian akan masuk ke lapisan konvolusi yang merupakan blok bangunan inti *CNN*, dimana sebagian besar komputasi dilakukan dilapisan ini. Hasil dari proses konvolusi di lapisan konvolusi kemudian akan diaktivasi dengan ReLU pada *activation layer* yang berguna untuk meningkatkan sifat non linearitas fungsi keputusan dan jaringan secara keseluruhan tanpa mempengaruhi bidang bidang reseptif pada *convolution layer*. *Pooling Layer* kemudian akan melakukan *down sampling* dari hasil konvolusi yang telah diaktivasi.

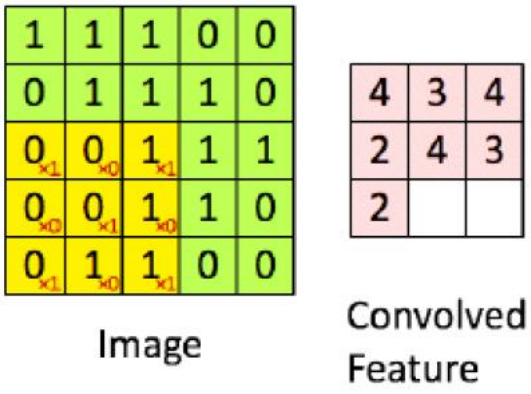
Nilai dari yang didapat dari *Pooling Layer* kemudian akan diubah menjadi *vector* yang dinamakan *flatten* yang kemudian akan masuk pada lapisan *Fully-Connected Layer* dan kemudian menghasilkan output dari hasil proses *feed forward* pada CNN[16].

2.4.1 *Feature Layer*

Feature layer terdiri dari 3 bagian, *convolution layer*, *activation layer* dan *pooling layer*.

2.4.1.1 *Convolution Layer*

Convolution Layer adalah lapisan yang didalamnya terdapat operasi konvolusi. Sebagian besar komputasi berada pada lapisan ini [16]. Lapisan ini juga yang menjadi proses utama yang mendasari sebuah CNN. Konvolusi merupakan sebuah istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel pada citra disemua *offset* memungkingkan. Kernel bergerak dari sudut kiri atas ke kanan bawah seperti diperoleh hasil konvolusi citra yang ditunjukkan pada gambar 2.2 berikut:



Gambar 2.2 Proses Konvolusi

Tujuan dilakukannya konvolusi adalah untuk mengekstraksi citra input yang dimana proses konvolusi akan menghasilkan tranformasi linear dari data input sesuai informasi spasial data. Secara formal operasi konvolusi dapat ditulis dengan rumus berikut:

$$s(t) = (x*w)(t)$$

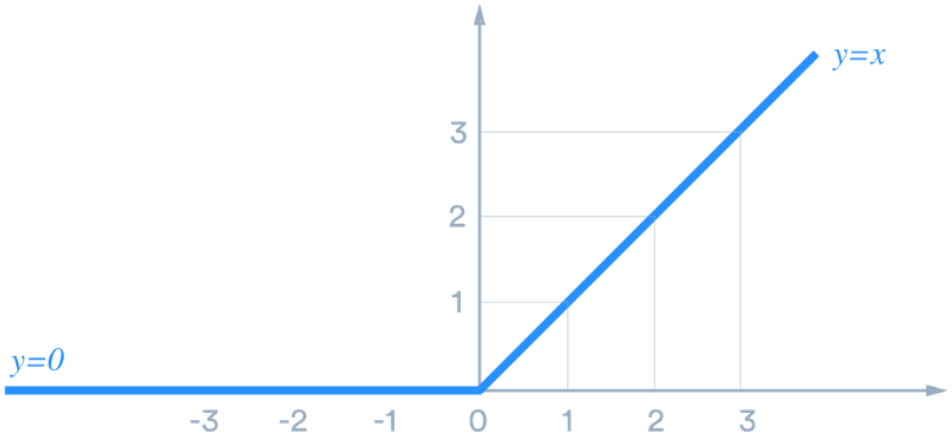
Fungsi *s(t)* memberikan output tunggal berupa *Feature Map*, argument pertama adalah input yang merupakan *x* dan argument kedua *w* sebagai filter sedangkan t sebagai pixel pada citra 2 dimensi yang akan diganti dengan i dan *j*[17].

2.4.1.2 *Activation Layer*

Fungsi aktivasi adalah fungsi non linear yang memungkinkan sebuah JST untuk dapat mentransformasikan data input menjadi dimensi yang lebih tinggi sehingga dapat dilakukan pemotongan *hyperlane* sederhana yang memungkinkan dilakukan klasifikasi. Pada penelitian ini, fungsi aktivasi yang digunakan yaitu fungsi *Rectified Linear Unit* (*ReLU*). Fungsi ini meningkatkan sifat nonlinearitas fungsi keputusan dan jaringan secara keseluruhan tanpa mempengaruhi bidang-bidang repesitif pada *convolution layer*. Fungsi *ReLU* mempunyai persamaan sebagai berikut.

$$f(x) = \max(0,x) \tag{2.1}$$

dimana: f(x) : fungsi *ReLU*
max(0,x) : fungsi maximum
x : nilai input



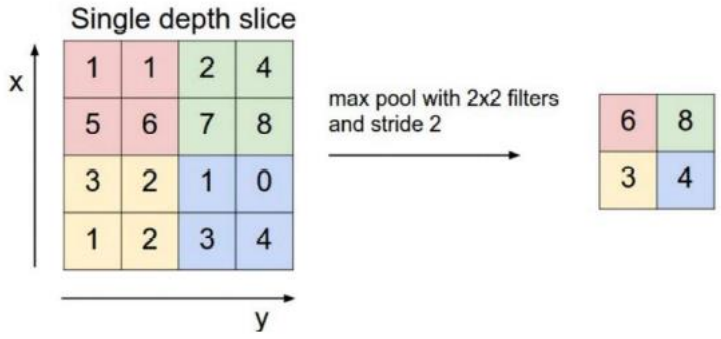
Gambar 2.3 Grafik Fungsi ReLU

Secara sederhana, cara kerja fungsi ReLU ialah dengan melihat nilai x yang masuk padanya. Ketika nilai x kurang dari atau sama dengan 0 maka nilai x = 0 sedangkan apabila nilai x lebih dari 0 maka nilai x = x. Bila digambarkan dengan persamaan maka akan persamaanya adalah sebagai berikut.

$$\max(0,x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{2.2}$$

2.4.1.3 *Pooling Layer*

Pooling layer adalah sebuah filter dengan ukuran dan *stride* tertentu yang bergeser pada seluruh area *feature map*. Pooling layer berfungsi menjaga ukuran data ketika proses *convolution*, yaitu dengan melakukan *downsampling* (pereduksian sampel). Pooling yang biasa digunakan adalah *Max Pooling* dan *Average Pooling*. Berikut contoh penerapan *max-pooling* untuk windows 2x2:



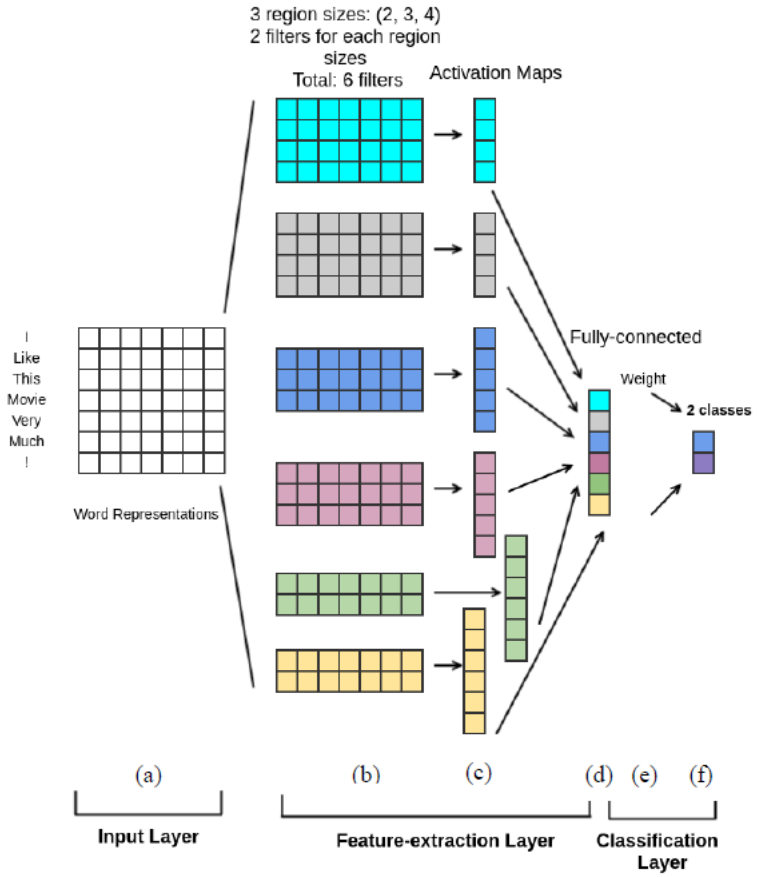
Gambar 2.4 *Pooling Layer*

2.4.2 *Fully-Connected Layer*

Pada *Fully-Connected layer* setiap *neurons* memiliki koneksi penuh ke semua aktivasi dalam lapisan sebelumnya. Hal ini sama persis dengan yang ada pada MLP. Model aktivasinya pun sama persis dengan yang ada di MLP, yaitu komputasi menggunakan suatu perkalian matriks yang diikuti dengan *bias* [15].

2.5 *Convolutional Neural Network* pada Text

Convolutional Neural Network adalah sebuah model yang banyak diaplikasikan pada pengenalan gambar, namun berbeda dengan NLP input yang digunakan adalah kalimat yang direpsentasikan sebagai matriks. Setiap baris dari matriks sesuai dengan satu token yang biasanya berbentuk kata. Setiap kata direpsentasikan dalam bentuk vector yang mewakili kata tersebut. Misalkan 1 kalimat mengandung 10 kata lalu menggunakan 5 dimensi embedding maka kita memiliki matriks 10x5 sebagai input. Berikut contoh visualisasi arsitektur CNN pada NLP dapat dilihat digambar 2.5

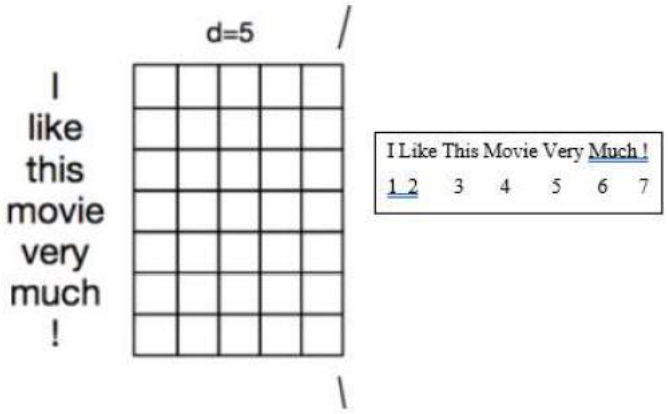


Gambar 2.5 Arsitektur CNN pada NLP [18]

Arsitektur tersebut merujuk pada penelitian yang telah dilakukan oleh Moch. Ari Nasichuddin dkk [1] dimana jumlah *filter* yang digunakan sebanyak 6 buah (2 *filter* ukuran 2x7, 2 *filter* ukuran 3x7 dan 2 *filter* ukuran 4x7).

2.5.1 *Input Layer*

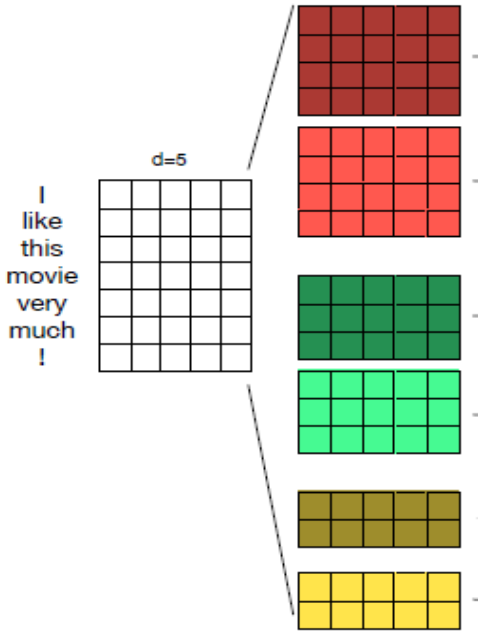
Langkah awal sebelum memasukan teks kedalam input layer CNN adalah menentukan berapa jumlah matriks yang dibutuhkan dalam kalimat teks tersebut yang nantinya akan diproses selanjutnya pada *Convolution Layer*. Berikut langkah langkah mengubah teks kedalam bentuk matriks dapat dilihat di gambar 2.6:



Gambar 2.6 Inputan CNN pada NLP [18]

2.5.2 *Convolution Layer*

Setelah matriks input layer sudah di dapatkan maka langkah selanjutnya adalah melakukan proses konvolusi pada *Convolution Layer* untuk mendapatkan nilai matriks yang baru dengan *hyperparameter filter*, *padding*, *stride* dll yang sudah ditentukan oleh sistem. Sebelum proses konvolusi dimulai dilakukan proses pemilahan setiap dari input layer lalu kemudian di filter setiap input layernya. *Stride* merupakan jumlah langkah pergeseran *filter*. Semakin kecil *stride* maka semakin detail informasi yang didapat, namun membutuhkan komputasi lebih berat dibanding jumlah *stride* yang lebih besar. Sedangkan *Padding* merupakan jumlah piksel yang akan ditambahkan pada setiap sisi dari matriks masukan. Biasanya *padding* berisi piksel yang nilainya 0. Tujuan dari *padding* adalah untuk mengurangi informasi yang terbuang sehingga ukuran matriks *input* dan *output* tetap sama.

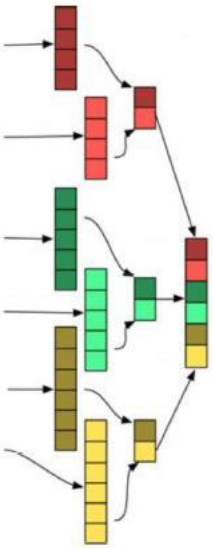


Gambar 2.7 Convolution Layer pada NLP [19]

2.5.3 *Max Pooling*

Max-Pooling adalah proses *downsampling* pereduksian sampel untuk menjaga ukuran data hasil proses konvolusi sebelumnya. *Max-Pooling* pada CNN untuk teks dilakukan dengan melakukan pengambilan nilai terbesar pada hasil dari proses konvolusi yang telah diaktivasi dengan *ReLU* sebelumnya [20]. Berikut persamaan *max-pooling* dengan {c} adalah nilai dari keseluruhan nilai *feature map*

$$c = \max\{c\} \tag{2.3}$$



Gambar 2.8 ***Max Pooling Layer dan Flatten***

Hasil dari *max-pooling* kemudian digabungkan menjadi flatten yang akan menjadi masukan untuk fully-connected layer.

2.5.4 *Fully Connected Layer*

Fully Connected Layer adalah *layer* yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Setiap *neurons* memiliki koneksi penuh ke semua aktivasi dalam lapisan sebelumnya.

Perbedaan antara *Fully Connected Layer* dengan *Convolution Layer* adalah neuron pada *Convolution Layer* terhubung hanya ke daerah tertentu pada input, sedangkan *Fully Connected Layer* memiliki neuron yang secara keseluruhan terhubung. *Fully Connected Layer* berperan untuk mengklasifikasi data masukan.

Sebelum masuk ke tahap *Fully Connected Layer*, *output* dari *layer* sebelumnya terlebih dahulu ditransformasikan menjadi bentuk *vektor* satu dimensi atau disebut sebagai proses *flatten*. Hasil dari *flatten* kemudian diklasifikasikan kedalam *output layer*. Berikut bentuk persamaan dari *output layer*:

a. *Output Layer*

$$M_i = \sum_{i=1}^{m} FC_j * W_{j,i} + B_{o,i} \quad (2.4)$$

dimana: M_i : masukkan untuk node dari flatten Z ke-i dengan jumlah node m

FC_j : node FC ke-j

$W_{j,i}$: bobot W untuk FC_j

$B_{o,i}$: bias W untuk M_i

2.5.5 *Softmax Classifier*

Softmax Classifier merupakan standar fungsi yang digunakan ketika proses klasifikasi melibatkan lebih dari dua kelas [21]. Bentuk persamaan *Softmax Classifier* adalah sebagai berikut.

$$Y_i = \frac{e^{y\_in_i}}{\sum_{i=1}^{m} e^{M}} \quad (2.5)$$

Keterangan: Y_i : keluaran untuk *output layer* ke-i

y_in_i : masukan untuk *node layer* ke-i

M : semua masukan untuk *output layer* sejumlah *m* buah

e : nilai 2.7182…

2.6 *Loss Function*

Loss Function merupakan fungsi untuk menghitung kerugian yang terkait dengan semua kemungkinan yang dihasilkan oleh suatu model. *Loss function* bekerja ketika model pembelajaran memberikan kesalahan yang harus diperhatikan. *Loss Function* yang baik adalah fungsi yang menghasilkan *error* seminimal mungkin. Pada penelitian ini *loss function* yang digunakan yaitu *Cross Entrophy Function*. Adapun persamaannya adalah sebagai berikut.

$$L = - \sum_{i}^{m} t_i \log(Y_i) \quad (2.6)$$

Keterangan: L : nilai *loss function*

t_i : nilai vektor yang diharapkan

m : jumlah kelas keluaran

Y_i : nilai keluaran ke-i sejumlah kelas *m* buah

2.7 *PHP*

Personal Home Page (PHP) adalah bahasa pemrograman yang digunakan untuk membuat Aplikasi Web. *PHP* dapat dijalankan pada sistem operasi apapun, seperti *Linux*, *Windows*, *OpenBSD*, *FreeBSD*, *MacOS*, *Solaris*, dan lain – lain. *PHP* disebut sebagai *server side scripting*, artinya skrip *PHP* dijalankan di sisi server, dimana setelah skrip *PHP* diolah di server, hasilnya dikirimkan ke *browser* (klien) [22].

2.8 *Imagemagick*

Imagemagick adalah *image utilities* yang bekerja dengan *interface command line*. Semua kebutuhan seperti mengonversi satu format ke format lain, menampilkan gambar, menampilkan identifikasi gambar dapat diperoleh dengan menggunakan *imagemagick* [23].

Karena *platform* pada penelitian ini menggunakan *Linux Ubuntu*, maka *imagemagick* yang digunakan pada penelitian ini tidak sama penggunaannya dengan *imagemagick* versi *windows*.

2.9 *Tesseract*

Tesseract merupakan *free engine Optical Character Recognition* (OCR) yang dirilis dibawah lisensi *Apache* dan pengembangannya disponsori oleh *Google*. *Tesseract* saat ini merupakan salah satu *engine* OCR *open source* yang paling akurat dibanding dengan *engine* yang lain. *Tesseract* dapat membaca berbagai format gambar dan mengkonversinya ke teks. Selain gambar, *Tesseract* juga dapat membaca file PDF [24].

OCR tool *Tesseract* digunakan pada penelitian ini karena mempunyai tingkat akurasi ekstraksi lebih besar dari 42% dan 25 kali lebih cepat untuk waktu pemrosesan ekstraksi dibandingkan dengan tool sebelumnya yaitu *OCRopus* [2].

2.10 *Smote*

Synthetic Minority Over-sampling Technique (*SMOTE*) merupakan metode yang populer diterapkan dalam rangka menangani ketidak seimbangan kelas. Metode *SMOTE* akan menambah jumlah data kelas minor agar setara dengan kelas mayor dengan cara membangkitkan data buatan [25].