BAB 3 PERANCANGAN DAN IMPLEMENTASI

3.1 Arsitektur Sistem

Pada penelitian ini akan dibuat program untuk mengklasifikasi *website* ke dalam *legitimate* atau *phishing*. *Website legitimate* adalah situs resmi yang digunakan dalam sehari-hari, dan *website phishing* yang adalah situs yang menyerupai situs resmi dengan tujuan mencuri data. Sistem ini melakukan klasifikasi kelas dengan menggunakan algoritme *Convolutional Neural Network* (CNN). Sistem akan menjalankan proses prediksi kategori URL melalui tiga tahap, yaitu :

1) Training

   Tahap *training* merupakan tahap untuk membangun model yang akan digunakan sebagai acuan dalam memprediksi URL. Melalui acuan model yang telah kita temukan, model tersebut nantinya akan digunakan sebagai proses dalam melakukan prediksi.

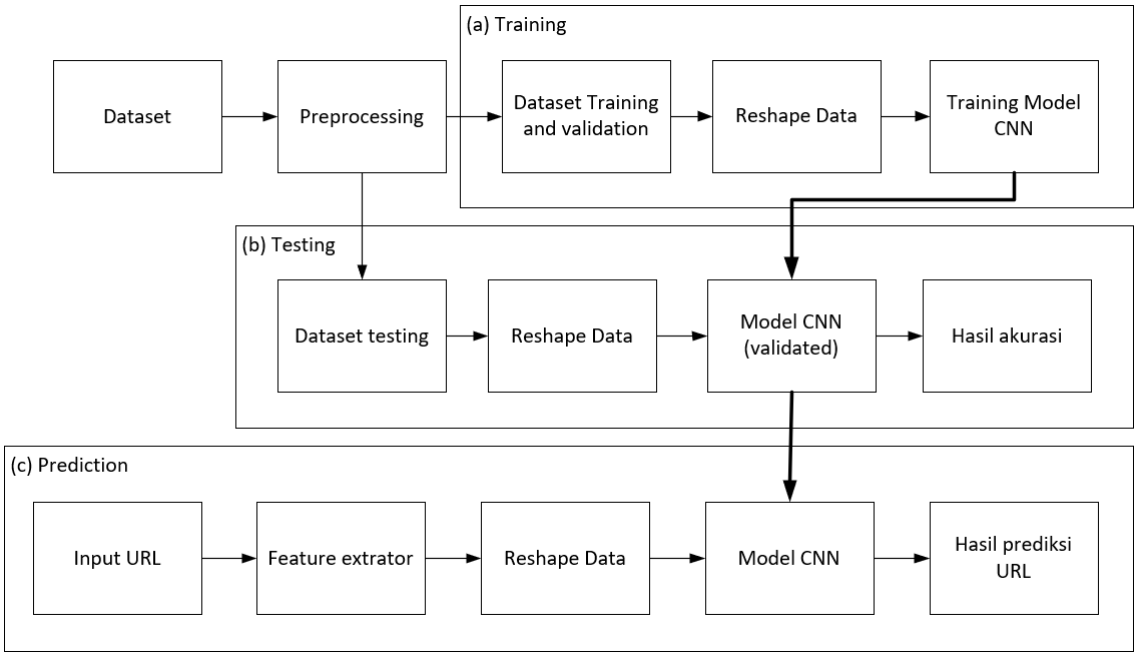
2) Testing

   Tahap *testing* merupakan tahap untuk menguji model yang sudah dibangun dengan menggunakan dataset testing. Melalui tahap *testing* bisa didapatkan epoch dan rasio yang terbaik.

3) Prediction

   Tahap *prediction* adalah tahap untuk memprediksi label URL dari *input* berdasarkan model yang sudah dibuat pada tahap training. Setelah melakukan tahap prediksi maka keluaran dari sistem akan menghasilkan simpulan akhir berupa kategori URL masuk dalam kategori *legitimate* atau *phishing*.

Proses utama dalam pembuatan model ini diawali dengan proses *training* kemudian proses *testing*. Proses ini bertujuan untuk pembentukan model yang akan digunakan untuk tahap *prediction*. Parameter untuk mengukur tingkat keberhasilan model adalah nilai *accuracy*. Nilai *accuracy* model dapat ditentukan dengan melakukan pengujian menggunakan dataset untuk *testing.* Proses tahapan desain sistem yang diusulkan diilustrasikan pada Gambar 3.1.

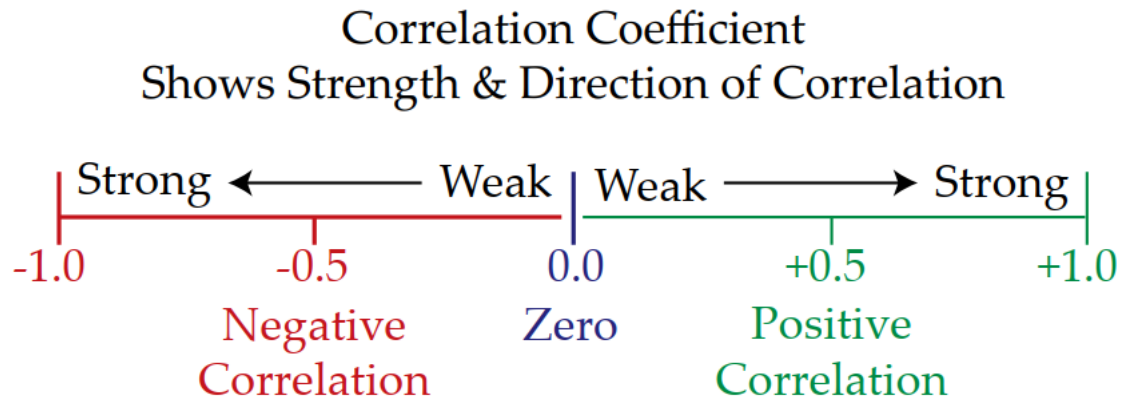


Gambar 3.1 Arsitektur Sistem

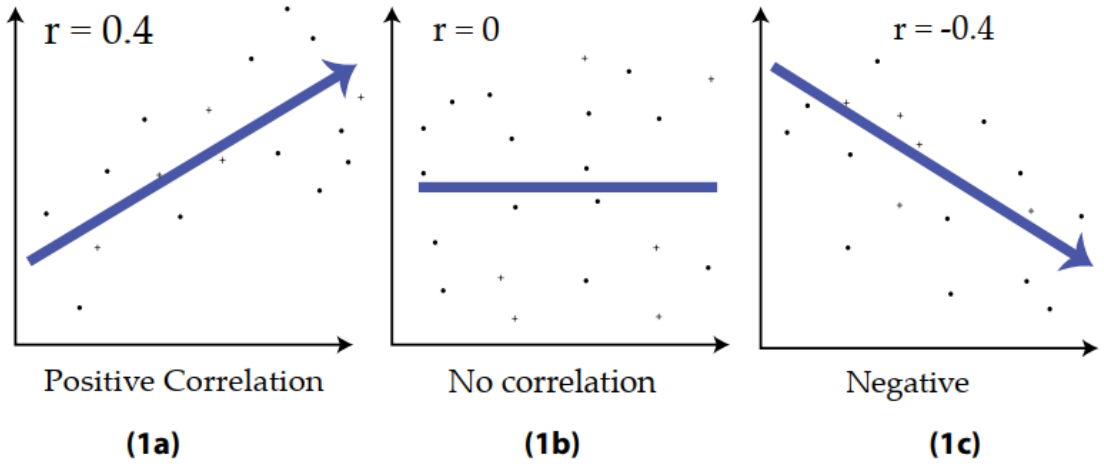
Setelah proses *training* dilakukan maka proses selanjutnya adalah proses *testing* kemudian yang terakhir proses *prediction*. Proses ini untuk input URL dari pengguna kemudian menggunakan model dari proses *training* untuk melakukan prediksi sebuah URL.

3.1.1 Dataset

Dataset yang digunakan berasal dari Mendeley, dataset awalnya ada 111 *feature.* Jumlah dataset yang digunakan adalah 88.647 dengan 58.000 website *legitimate* dan 30.647 website *phishing* [6]. Dataset mengalami proses pengurangan *feature* dengan melihat *spearman correlation* dari *software orange* [7]. Pengurangan *feature* melihat dari *correlation* yang kuat yaitu lebih besar dari 0,5 pada *positive correlation* atau lebih kecil dari -0,5 pada *negative correlation* sehingga hanya menjadi hanya 39 *feature* [16].



Gambar 3.2 Spektrum Koefisien Korelasi (-1 hingga +1) [16]



Gambar 3.3 *Scatter Plot* Menunjukkan Korelasi antara Dua Variabel [16]

Gambar 3.4 Pemisahan URL ke *Sub-Strings* [6]

Tabel 3.1 adalah penjelasan dataset yang dipakai setelah mengalami pengurangan *feature*.

Tabel 3.1 Penjelasan Dataset Setelah mengalami Pengurangan *Feature*

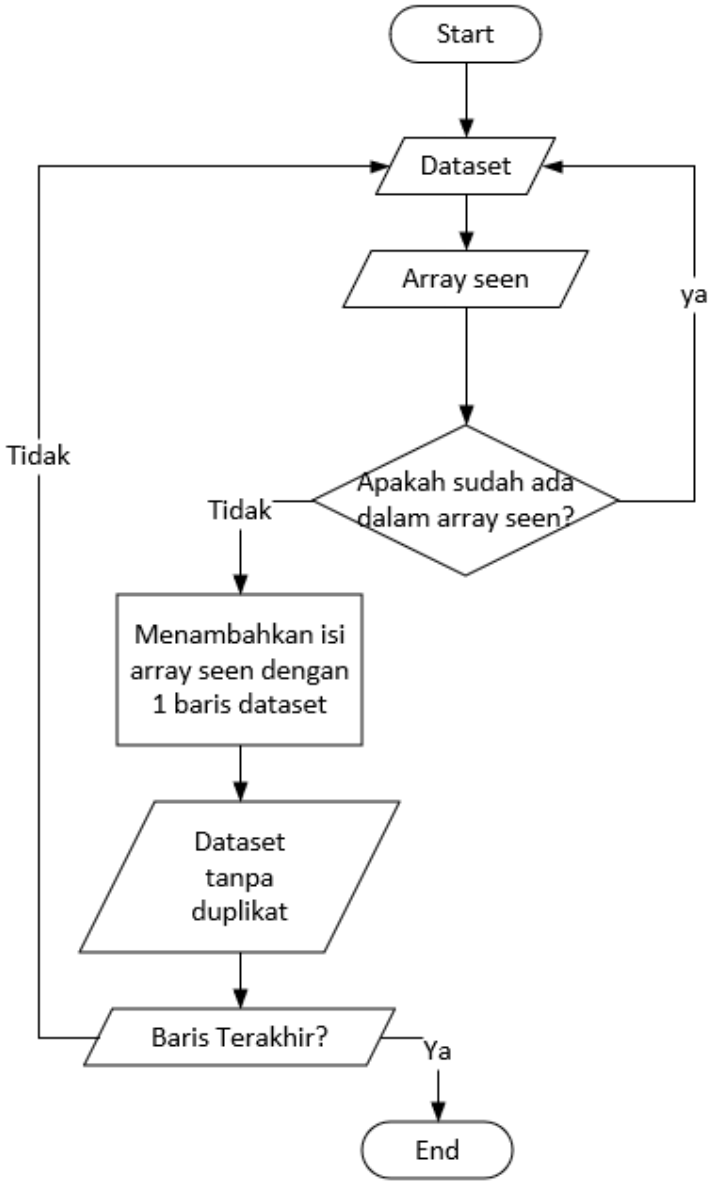
No	Fitur	Format	Description	Values
1	qty_slash_url	Jumlah ”/” di URL	Numeric	
2	length_url	Jumlah karakter di URL	Numeric	
3	qty_dot_directory	Jumlah ”.” di directory pada URL	Numeric	
4	qty_hyphen_directory	Jumlah ”-” di directory pada URL	Numeric	
5	qty_underline_directory	Jumlah ”_” di directory pada URL	Numeric	
6	qty_slash_directory	Jumlah ”/” di directory pada URL	Numeric	
7	qty_questionmark_directory	Jumlah ”?” di directory pada URL	Numeric	
8	qty_equal_directory	Jumlah ”=” di directory pada URL	Numeric	

No	Fitur	Format	Description	Values
9	qty_at_directory	Jumlah ”@” di directory pada URL	Numeric	
10	qty_and_directory	Jumlah ”&” di directory pada URL	Numeric	
11	qty_exclamation_directory	Jumlah ”!” di directory pada URL	Numeric	
12	qty_space_directory	Jumlah ” ” di directory pada URL	Numeric	
13	qty_tilde_directory	Jumlah ” di directory pada URL	Numeric	
14	qty_comma_directory	Jumlah ”,” di directory pada URL	Numeric	
15	qty_plus_directory	Jumlah ”+” di directory pada URL	Numeric	
16	qty_asterisk_directory	Jumlah ”*” di directory pada URL	Numeric	
17	qty_hashtag_directory	Jumlah ”#” di directory pada URL	Numeric	
18	qty_dollar_directory	Jumlah ”$” di directory pada URL	Numeric	
19	qty_percent_directory	Jumlah ”%” di directory pada URL	Numeric	
20	directory_length	Jumlah karakter di directory pada URL	Numeric	
21	qty_dot_file	Jumlah ”.” di file pada URL	Numeric	
22	qty_hyphen_file	Jumlah ”-” di file pada URL	Numeric	
23	qty_underline_file	Jumlah ”_” di file pada URL	Numeric	
24	qty_slash_file	Jumlah ”/” di file pada URL	Numeric	
25	qty_questionmark_file	Jumlah ”?” di file pada URL	Numeric	
26	qty_equal_file	Jumlah ”=” di file pada URL	Numeric	
27	qty_at_file	Jumlah ”@” di file	Numeric	

No	Fitur	Format	Description	Values
		pada URL		
28	qty_and_file	Jumlah ”&” di file pada URL	Numeric	
29	qty_exclamation_file	Jumlah ”!” di file pada URL	Numeric	
30	qty_space_file	Jumlah ” ” di file pada URL	Numeric	
31	qty_tilde_file	Jumlah ” di file pada URL	Numeric	
32	qty_comma_file	Jumlah ”,” di file pada URL	Numeric	
33	qty_plus_file	Jumlah ”+” di file pada URL	Numeric	
34	qty_asterisk_file	Jumlah ”*” di file pada URL	Numeric	
35	qty_hashtag_file	Jumlah ”#” di file pada URL	Numeric	
36	qty_dollar_file	Jumlah ”$” di file pada URL	Numeric	
37	qty_percent_file	Jumlah ”%” di file pada URL	Numeric	
38	file_length	Jumlah karakter di file pada URL	Numeric	
39	phishing	Apakah ini situs phishing?	Boolean	[0, 1]

3.1.2 Training

Tahap *training* merupakan tahap untuk membangun model yang akan digunakan sebagai acuan dalam memprediksi URL. Melalui acuan model yang telah kita temukan, model tersebut nantinya akan digunakan sebagai proses dalam melakukan prediksi. Tahap *training* diawali tahap *preprocessing* akan menghapus data duplikat pada data berupa csv sehingga data yang pada csv keseluruhannya adalah data yang berbeda. Gambar 3.5 dan Algoritma 3.1 akan menjelaskan algoritme dari *preprocessing* [8]. Setelah mengalami proses *preprocessing* jumlah dataset yang digunakan menjadi 19868 dengan 2984 website *legitimate* dan 16884 website *phishing*.



Gambar 3.5 *Flowchart Preprocessing*

Algoritma 3.1: Algoritma untuk tahap-tahap *preprocessing*

Masukan: Data dari dataset

Keluaran: Data yang sudah tidak ada duplikat

Proses:

1: Memuat dataset

2: Menambahkan satu baris pada dataset ke array seen bila baris pada dataset belum ada pada array seen

3: Bila baris belum berakhir ulangi langkah 2 pada baris baru

4: Hasil dari array seen dibuat data baru yang sudah tidak ada duplikat

Setelah tahap *preprocessing* maka akan ditentukan rasio dataset berupa train_size, valid_size, test_size akan di tentukan. *Training model* adalah sebuah proses dimana algoritme *machine learning* data pelatihan untuk dipelajari, besaran data yang digunakan akan ditentukan pada train_size. *Validation data* adalah set data, terpisah dari set pelatihan, yang digunakan untuk menvalidasi kinerja model besarannya ditentukan pada valid_size. *Testing data* adalah set data yang akan digunakan untuk menilai kinerja model yang besarannya ditentukan pada test_size.

Reshape data adalah fungsi *python* yang digunakan untuk memberikan bentuk baru ke array tanpa mengubah datanya. *Reshape* membutuhkan 3 variable yaitu jumlah sample training, jumlah feature berjumlah tiga puluh delapan, dan dimensi berjumlah satu. Pada tahap *Reshape array, array* yang dihasilkan dari dataset diubah bentuknya. *Reshape* digunakan supaya model dapat membaca data yang sudah di inputkan. Pelatihan model diawali dengan penentuan parameter dari model CNN yaitu *epoch* yang menunjukkan jumlah berapa kali *machine learning* akan bekerja melalui seluruh dataset pelatihan.

3.1.3 Model *Convolutional Neural Network*

Bagian ini menjelaskan model CNN (*Convolutional Neural Network*). Dalam CNN ada proses konvolusi yang berada pada *convolution layer*, lapisan yang melakukan proses konvolusi. *Convolution layer* terdiri dari neuron yang membentuk sebuah filter dengan panjang dan tinggi tertentu. Filter inilah yang akan bergerak secara beraturan untuk mengekstrak informasi yang dapat diperoleh. Algoritma konvolusi ada pada Algoritma 3.2 dan contoh proses konvolusi ada pada Gambar 3.6.

Algoritma 3.2: Algoritma untuk proses konvolusi

Masukan: *Input dari dataset*, Panjang *filter*

Keluaran: Hasil konvolusi

Proses:

1: *Input dari dataset* ke 1 dikali *filter* ke 1

2: *Input dari dataset* ke 2 dikali *filter* ke 2

3: Langkah 1 dan 2 diulangi sampai pada *input* ke n dikali *filter* n

4: *Filter* digeser 1 langkah pada *input dari dataset*

5:Langkah 1-4 diualngi sampai *filter* sudah sampai pada ujung dari *input*

Proses konvolusi membutuhkan beberapa parameter, yaitu *input* $x(..)$, parameter n yang yang menentukan banyaknya input, parameter k yang menentukan panjang kernel, dan parameter s yaitu *stride* yang akan menentukan pergeseran filter. *Stride* disini artinya jumlah pergeseran kernel terhadap matriks input berjumlah satu.

Perhitungan jumlah *output* ada pada Persamaan 2.3 dan perhitungan proses konvolusi pada *input* yaitu [1 25 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 8 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7] dengan n = 38, k = 7 dan s = 1

$$o = \left[\frac{n-k}{s}\right] + 1$$

$$o = \left[\frac{38-7}{1}\right] + 1 = 32$$

$$y(n)\begin{cases}\sum_{i=0}^{k} x(n+i)h(i), & if\ n = 0\\ \sum_{i=0}^{k} x(n+i+(s-1))h(i), & otherwise\end{cases}$$

$$y(0) = x(0)h(0) + x(1)h(1) + x(2)h(2) + x(3)h(3) + x(4)h(4) + x(5)h(5) + x(6)h(6)$$

$$y(0) = 1 \times 1 + 25 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 + 1 \times 1 + 0 \times 1 = 28$$

$$y(1) = x(1)h(0) + x(2)h(1) + x(3)h(2) + x(4)h(3) + x(5)h(4) + x(6)h(5) + x(7)h(6)$$

$$y(1) = 25 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 = 27$$

$$y(2) = x(2)h(0) + x(3)h(1) + x(4)h(2) + x(5)h(3) + x(6)h(4) + x(7)h(5) + x(8)h(6)$$

$$y(2) = 1 \times 1 + 0 \times 1 + 0 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 2$$

$$y(3) = x(3)h(0) + x(4)h(1) + x(5)h(2) + x(6)h(3) + x(7)h(4) + x(8)h(5) + x(9)h(6)$$

$$y(3) = 1 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 1$$

$$y(4) = x(4)h(0) + x(5)h(1) + x(6)h(2) + x(7)h(3) + x(8)h(4) + x(9)h(5) + x(10)h(6)$$

$$y(4) = 0 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 1$$

$$y(5) = x(5)h(0) + x(6)h(1) + x(7)h(2) + x(8)h(3) + x(9)h(4) + x(10)h(5) + x(11)h(6)$$

$$y(5) = 1 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 1$$

$$y(6) = x(6)h(0) + x(7)h(1) + x(8)h(2) + x(9)h(3) + x(10)h(4) + x(11)h(5) + x(12)h(6)$$

$$y(6) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(7) = x(7)h(0) + x(8)h(1) + x(9)h(2) + x(10)h(3) + x(11)h(4) + x(12)h(5) + x(13)h(6)$$

$$y(7) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(8) = x(8)h(0) + x(9)h(1) + x(10)h(2) + x(11)h(3) + x(12)h(4) + x(13)h(5) + x(14)h(6)$$

$$y(8) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(9) = x(9)h(0) + x(10)h(1) + x(11)h(2) + x(12)h(3) + x(13)h(4) + x(14)h(5) + x(15)h(6)$$

$$y(9) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(10) = x(10)h(0) + x(11)h(1) + x(12)h(2) + x(13)h(3) + x(14)h(4) + x(15)h(5) + x(16)h(6)$$

$$y(10) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(11) = x(11)h(0) + x(12)h(1) + x(13)h(2) + x(14)h(3) + x(15)h(4) + x(16)h(5) + x(17)h(6)$$

$$y(11) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(12) = x(12)h(0) + x(13)h(1) + x(14)h(2) + x(15)h(3) + x(16)h(4) + x(17)h(5) + x(18)h(6)$$

$$y(12) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(13) = x(13)h(0) + x(14)h(1) + x(15)h(2) + x(16)h(3) + x(17)h(4) + x(18)h(5) + x(19)h(6)$$

$$y(13) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 8 \times 1 = 8$$

$$y(14) = x(14)h(0) + x(15)h(1) + x(16)h(2) + x(17)h(3) + x(18)h(4) + x(19)h(5) + x(20)h(6)$$

$$y(14) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 8 \times 1 + 1 \times 1 = 9$$

$$y(15) = x(15)h(0) + x(16)h(1) + x(17)h(2) + x(18)h(3) + x(19)h(4) + x(20)h(5) + x(21)h(6)$$

$$y(15) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 8 \times 1 + 1 \times 1 + 0 \times 1 = 9$$

$$y(16) = x(16)h(0) + x(17)h(1) + x(18)h(2) + x(19)h(3) + x(20)h(4) + x(21)h(5) + x(22)h(6)$$

$$y(16) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 8 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 = 9$$

$$y(17) = x(17)h(0) + x(18)h(1) + x(19)h(2) + x(20)h(3) + x(21)h(4) + x(22)h(5) + x(23)h(6)$$

$$y(17) = 0 \times 1 + 0 \times 1 + 8 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 9$$

$$y(18) = x(18)h(0) + x(19)h(1) + x(20)h(2) + x(21)h(3) + x(22)h(4) + x(23)h(5) + x(24)h(6)$$

$$y(18) = 0 \times 1 + 8 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 9$$

$$y(19) = x(19)h(0) + x(20)h(1) + x(21)h(2) + x(22)h(3) + x(23)h(4) + x(24)h(5) + x(25)h(6)$$

$$y(19) = 8 \times 1 + 1 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 9$$

$$y(20) = x(20)h(0) + x(21)h(1) + x(22)h(2) + x(23)h(3) + x(24)h(4) + x(25)h(5) + x(26)h(6)$$

$$y(20) = 1 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 1$$

$$y(21) = x(21)h(0) + x(22)h(1) + x(23)h(2) + x(24)h(3) + x(25)h(4) + x(26)h(5) + x(27)h(6)$$

$$y(21) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(22) = x(22)h(0) + x(23)h(1) + x(24)h(2) + x(25)h(3) + x(26)h(4) + x(27)h(5) + x(28)h(6)$$

$$y(22) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(23) = x(23)h(0) + x(24)h(1) + x(25)h(2) + x(26)h(3) + x(27)h(4) + x(28)h(5) + x(29)h(6)$$

$$y(23) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(24) = x(24)h(0) + x(25)h(1) + x(26)h(2) + x(27)h(3) + x(28)h(4) + x(29)h(5) + x(30)h(6)$$

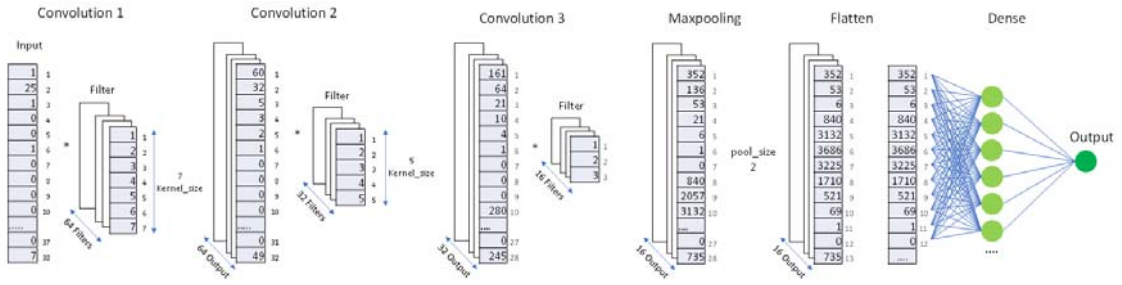
$$y(24) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(25) = x(25)h(0) + x(26)h(1) + x(27)h(2) + x(28)h(3) + x(29)h(4) + x(30)h(5) + x(31)h(6)$$

$$y(25) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$

$$y(26) = x(26)h(0) + x(27)h(1) + x(28)h(2) + x(29)h(3) + x(30)h(4) + x(31)h(5) + x(32)h(6)$$
$$y(26) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$
$$y(27) = x(27)h(0) + x(28)h(1) + x(29)h(2) + x(30)h(3) + x(31)h(4) + x(32)h(5) + x(33)h(6)$$
$$y(27) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$
$$y(28) = x(28)h(0) + x(29)h(1) + x(30)h(2) + x(31)h(3) + x(32)h(4) + x(33)h(5) + x(34)h(6)$$
$$y(28) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$
$$y(29) = x(29)h(0) + x(30)h(1) + x(31)h(2) + x(32)h(3) + x(33)h(4) + x(34)h(5) + x(35)h(6)$$
$$y(29) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$
$$y(30) = x(30)h(0) + x(31)h(1) + x(32)h(2) + x(33)h(3) + x(34)h(4) + x(35)h(5) + x(36)h(6)$$
$$y(30) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 = 0$$
$$y(31) = x(31)h(0) + x(32)h(1) + x(33)h(2) + x(34)h(3) + x(35)h(4) + x(36)h(5) + x(37)h(6)$$
$$y(31) = 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 0 \times 1 + 7 \times 1 = 7$$

Kemudian proses konvolusi dilakukan 2 kali lagi, kemudian dilakukan proses *max pooling*. *Max pooling* akan memilih atau cukup mengambil nilai maksimum sesuai dengan namanya. Penggunaan *pooling layer* bertujuan mengurangi dimensi dari *feature map (downsampling)*, sehingga mempercepat komputasi karena parameter yang harus diproses semakin sedikit. Jika tahap sebelumnya kita sudah melakukan *polling*, selanjutnya kita masuk ke tahap *flattening* yaitu merubah dari matriks yang ada di *pooling layer* menjadi satu kolom saja yang dapat diartikan menjadi sebuah vektor tunggal [12].



Gambar 3.6 Skema *Convolution,Max Pooling, Flatten* dan *Dense*

Tahap selanjutnya adalah *dense layer* adalah *neural network* yang terhubung secara mendalam, yang berarti setiap *neuron* pada lapisan padat menerima input dari semua neuron pada lapisan sebelumnya. Algoritma 3.3 akan menjelaskan cara kerja *dense layer*. *Dense layer* ditemukan menjadi lapisan yang paling umum digunakan dalam model [17].

Algoritma 3.3: Algoritma untuk *dense*

Masukan: hasil dari *flatten*, label pada dataset

Keluaran: *bias* dan *weight*

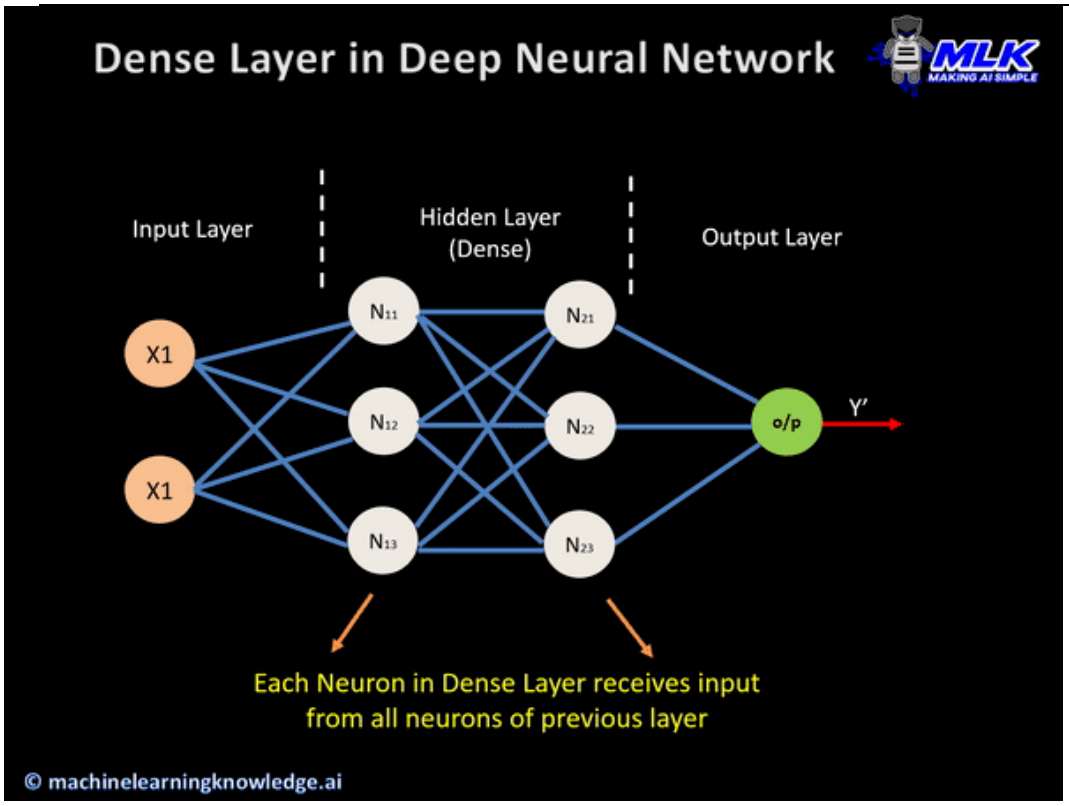
Proses:

1: hasil dari *flatten* dibandingkan dengan label pada dataset

2: langkah 1 diulangi untuk semua hasil flatten juga label pada dataset

3: didapatkan *bias* dan *weight* dari pembelajaran untuk *dense layer*

4: Hasil dari pembelajaran digunakan untuk melakukan prediksi



Gambar 3.7 *Dense Layer* [17]

Tabel 3.2 Pemetaan Tahapan Proses Arsitektur Model

Tahap	Size	Total parameter
Input	(None, 38, 1)	0

Tahap	Size	Total parameter
Conv1D_1 (Conv1D)	(None, 32, 64)	512
Conv1D_2 (Conv1D)	(None, 28, 32)	10272
Conv1D_3 (Conv1D)	(None, 26, 16)	1552
MaxPooling1D (MaxPooling1D)	(None, 13, 16)	0
flatten (Flatten)	(None, 208)	0
Dense_1 (Dense)	(None, 32)	6688
Dense_2 (Dense)	(None, 1)	33
Total parameter		19057

3.1.4 Penilaian Training Model *Convolutional Neural Network*

Penilaian *training* akan menggunakan *confusion matrix* yang berisi hasil prediksi dengan data yang sebenarnya. Dalam penelitian ini ada *predicted negative* yaitu hasil dari yang diprediksikan negative, *predicted positive* yaitu dari hasil yang diprediksikan positif, *actual negative* yaitu data yang sebenarnya negatif, *actual positive* yaitu data yang sebenarnya positif. Tabel 3.3 merupakan contoh dari *confusion matrix*.

Tabel 3.3 Contoh *Confusion Matrix*

	Predicted Negative	***Predicted Positive***
Actual Negative	60	258
Actual Positive	19	1650

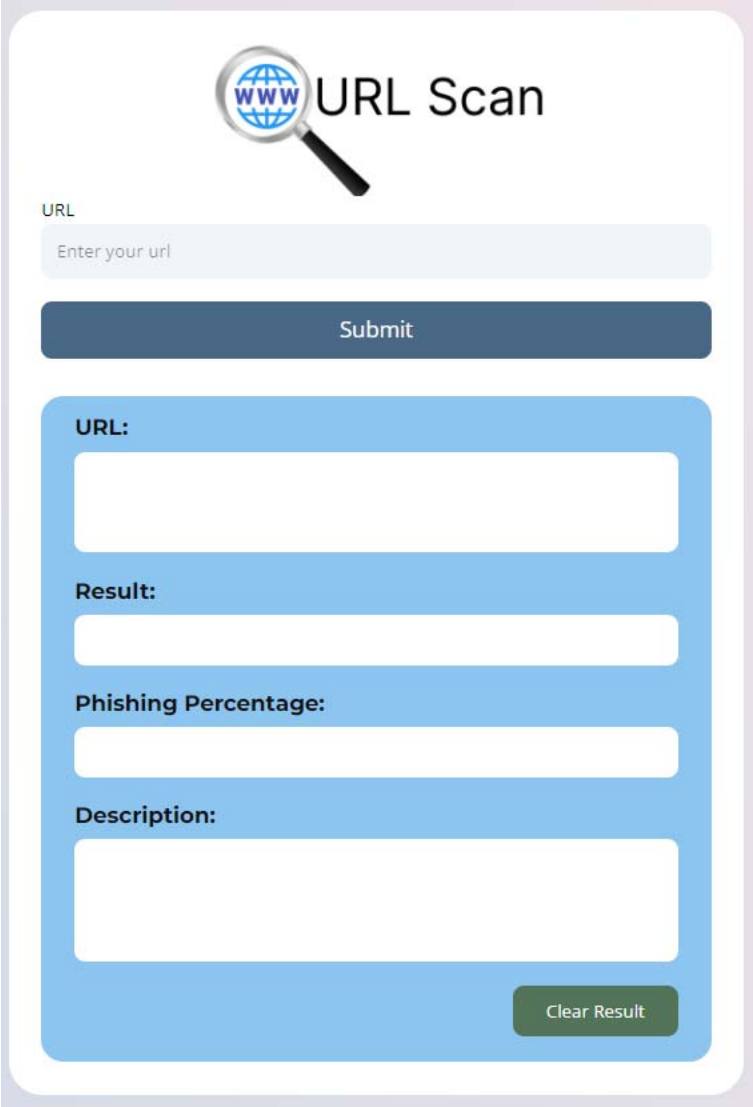
Nilai *true positive* (TP) didapatkan jika data aktual dan prediksi data bernilai positif. Nilai *true negative* (TN) didapatkan jika data aktual dan prediksi data bernilai negatif. Nilai *false positive* (FP) didapatkan jika data aktual bernilai negatif namun prediksi data bernilai positif. Nilai *false negative* (FN) didapatkan jika data aktual bernilai positif namun prediksi data bernilai negatif. Dibawah berikut adalah contoh perhitungan nilai *accuracy* berdasarkan Persamaan 2.5 dan komponen-komponen pada Tabel 3.3.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{1650 + 60}{1650 + 60 + 258 + 19} = \frac{1710}{1987} \approx 0{,}860594$$

3.1.5 Implementasi Sistem

Untuk mempermudah pengguna dalam menggunakan sistem ini maka dibuatlah *user interface* sederhana pada Gambar 3.4 yang mempermudah dalam menggunakan sistem untuk melakukan prediksi. *Library* yang digunakan adalah *Flask.*



Gambar 3.8 *User Interface* Sistem [18]

Input berupa URL yang misalnya: https://www.youtube.com/ kemudian *feature extrator* adalah proses dari URL yang dimasukan menjadi *feature* di *machine learning* dari *feature extractor*. Reshape data adalah fungsi *python* yang digunakan untuk memberikan bentuk baru ke array tanpa mengubah datanya. Array dari *feature extrator* diubah menjadi bentuk yang dapat dibaca oleh CNN (*Convolutional Neural Network*). Model CNN diambil

dari hasil pelatihan model untuk melakukan prediksi, hasil prediksi ditampilkan dari *user interface* pada sistem.