BAB II

LANDASAN TEORI

2.1. *State Of The Art*

2.1.1. Studi Literatur

Jurnal yang berjudul *Smart Home Security* Menggunakan *Face Recognition* Dengan Metode *Eigenface* Berbasis *Raspberry Pi* (Rudi Kurniawan & Zulius, 2019) meneliti tentang keamanan rumah, maksud dari penelitian tersebut adalah untuk mengidentifikasi sebuah sistem baru yang berfungsi untuk mencegah tindak pembobolan dan pencurian rumah karena lemahnya tingkat pengaman kunci atau gembok, sedangkan tujuan yang ingin dicapai adalah agar dapat membantu para pemilik rumah supaya dapat mengawasi rumah mereka ketika mereka tidak ada. Metode yang digunakan untuk menyelesaikan masalah yaitu metode *eigenface* berbasis raspberry pi. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu membahas tentang keamanan rumah berdasarkan *face recognition*. Kesimpulan dari penelitian tersebut adalah alat dapat mengenali citra wajah dengan tingkat keberhasilan sampai 90% pada jarak 25 cm dengan rata-rata keberhasilan sebesar 72.5 %.

Jurnal yang berjudul *Internet Of Things* Sistem Keamanan Rumah Berbasis *Raspberry Pi* Dan Telegram Messenger (Kurniawan et al., 2018). Maksud dari penelitian tersebut adalah bagaimana *Internet of Things* mampu melakukan monitoring rumah dari jarak jauh dengan memanfaatkan aplikasi *instant messenger* telegram. Penggunaan telegram *messenger* pada penelitian tersebut karena sifatnya

yang *open source*. Metode yang digunakan untuk menyelesaikan masalah yaitu menggunakan metode pengkuran *delay* menggunakan *stopwatch*. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu membahas tentang keamanan rumah dengan memanfaatkan telegram. Kesimpulan dari penelitain tersebut adalah Jarak maksimal sensor PIR dapat mendeteksi adanya suatu pergerakan obyek adalah 6 meter dengan Sudut sensor arah horizontal sebesar 90° sampai dengan 135° dan posisi arah vertikal sebesar 60° sampai dengan 120°.

Jurnal yang berjudul Sistem Pengontrolan Dan Keamanan Rumah Pintar *(Smart Home)* Berbasis *Android.* (Putri & Yendri, 2018) meneliti tentang pengontrolan dan keamanan rumah, maksud dari penelitian tersebut adalah bertujuan untuk meningkatkan keamanan, efisiensi dan kenyamanan penghuninya. Metode yang digunakan untuk menyelesaikan masalah yaitu metode penelitian studi literatur dan penelitian eksperimen. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu membahas tentang keamanan rumah. Kesimpulan dari penelitian tersebut adalah sistem dapat menghidupkan dan mematikan lampu melalui *smart phone* dengan tingkat keberhasilan 100%

Jurnal yang berjudul *Detektor* Keamanan Rumah Melalui *Telegram Messeger*. (Yuliza, 2018) meneliti tentang mengimplementasian pada pengendalian perangkat rumah dan memberikan keamanan ketika pengguna jauh dari tempat. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu membahas tentang keamanan rumah dengan memanfaatkan

telegram. kesimpulan dari penelitian tersebut adalah *sensor PIR* dapat mendeteksi dan mengirimkan pesan ke *telegram messenger*.

Jurnal yang berjudul Sistem Pendeteksi Ancaman Keamanan Rumah dengan Menggunakan Telegram Berbasis *Internet Of Things*. (Eka Prasetiyo & Setiyadi, 2021) meneliti tentang *prototype* sistem pendeteksi ancaman keamanan dengan menggunakan telegram berbasis *internet of things*. Sistem ini dapat meminimalisir terjadinya ketidaknyamanan pemilik rumah saat tidak berada di dalam rumah. Metode penelitian yang digunakan pada penelitian ini adalah metode *prototyping*. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu membahas tentang keamanan rumah berbasis *IOT*. Kesimpulan dari penelitain tersebut adalah Sistem dapat mengirim notifikasi kepada pemilik rumah sebagai tanda bahwa sistem mendeteksi adanya kebakaran atau penyusup masuk kedalam rumah yang sedang ditinggalkan.

Jurnal yang berjudul Rancang bangun Sistem Keamanan Rumah Berbasis *Internet Of Things Dengan Platform Android.* (Khana & Uus Usnul, 2018) meneliti tentang sistem keamanan rumah basis *Internet of Things* dengan memanfaatkan sensor *Passive Infra Red (PIR)* untuk mendeteksi adanya objek yang bergerak dan *sensor MQ-2* untuk mendeteksi adanya kebocoran gas serta melakukan kontrol terhadap beberapa *device* yang berhubungan dengan *system* keamanan rumah seperti lampu dan *solenoid door lock* untuk mengunci pintu. Metode yang digunakan untuk menyelesaikan masalah yaitu *motion detection*. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu membahas tentang keamanan rumah berbasis *IOT*. Kesimpulan dari penelitian

tersebut adalah Sistem keamanan rumah berbasis Internet of Things dengan platform android menggunakan mikrokontroler arduino yang dikombinasikan dengan sensor PIR, sensor MQ-2, RFID dan lampu telah berfungsi dengan baik sedangkan untuk proses pengiriman notifikasi email kepada pemilik rumah tergantung dari kondisi jaringan internet dan provider operator seluler yang digunakan.

Jurnal yang berjudul Pengenalan Wajah dengan *Algoritma Support Vector Machine* dan *Sobel Edge Detection* Berbasis *Computer Vision* dan *Cafe Framework.* (Hidayatulloh et al., 2020) meneliti tentang menerapkan *algoritma sup-port vector machine* dan *ekstraksi fitur edge detection*. Metode yang digunakan yaitu dengan *metode sobel* dalam melakukan pengenalan wajah. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu membahas tentang *face recognition*. Kesimpulan dari penelitian tersebut adalah berdasarkan pengujian dengan tingkat persentase keberhasilan seluruh sistem mencapai 100%. Menginput citra baru, *accuracy* dengan jumlah data latih 30, 40 dan 50 data setiap kelas sebesar 89% lebih tinggi dari jumlah data latih 10 dan 20 data dengan *accuracy* 70%.

Jurnal yang berjudul Rancang Bangun *New Normal Covid-19* Masker *Detektor* Dengan Notifikasi Telegram Berbasis *Internet Of Things*. (Lambacing & Ferdiansyah, 2020) meneliti tentang bagaimana perusahaan dapat mendisiplinkan karyawannya untuk menggunakan masker sebelum masuk ke kantor, maka dibuatlah *New Normal COVID-19* Masker *Detektor* dengan *Notifikasi Telegram* berbasis *Internet Of Things*. Pembaruan dari penelitian tersebut yang dianggap

relevan dengan penelitian yang dilakukan yaitu meneliti tentang *IOT* dengan memanfaatkan telegram. Kesimpulan dalam penelitian tersebut yaitu sistem mampu mendeteksi masker yang digunakan oleh manusia juga dapat mengirimkan notifikasi langsung ke telegram sebagai pemberitahuan ke petugas keamanan.

Jurnal yang berjudul Perancangan Sistem Pengontrol Keamanan Rumah Dengan *Smart CCTV* Menggunakan *Arduino* Berbasis Telegram (Setiawan et al., 2019) meneliti tentang perancangan alat *smart CCTV* menggunakan *arduino* yang dapat berfungsi sebagai alat untuk sistem keamanan rumah. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang kemanan rumah dengan memanfaatkan telegram. Kesimpulan dalam penelitian tersebut yaitu alat *smart CCTV* menggunakan arduino akan mudah di pantau dimanapun pemilik rumah berada, sehingga bisa merasa aman ketika rumah dalam keadaan kosong saat bepergian.

Jurnal yang berjudul Sistem Monitoring Keamanan Gedung berbasis *Rasberry Pi.* (Sirait, 2016) meneliti tentang sistem monitoring keamanan gedung yang dilengkapi dengan *sensor passive infra red (HC-SR501)* yang dipasang diatas pintu utama gedung serta kamera *webcam*, kemudian sebagai pemrosesnya digunakan *minikomputer raspberry pi*. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang kemanan dengan memanfaatkan *instan messenger* sebagai *notifikasi*. Kesimpulan dari penelitian tersebut adalah Setiap ada pergerakan dalam radius sensor, maka Raspberry Pi akan mengirim pesan teks dan gambar (foto) kepada user *via whatsapp messenger*.

Jurnal yang berjudul Implementasi *Raspberry Pi* Untuk Rancang Bangun sistem Keamanan Pintu Ruang Server Dengan Pengenalan Wajah Menggunakan *Metode Triangle Face.* (Polinema et al., 2017) meneliti tentang sistem keamanan ruang *server* yang mudah untuk diaplikasikan dan murah dalam segi biaya pembuatan dan perawatan serta berteknologi, mengingat pentingnya keamanan data dan informasi yang tersimpan dalam *server* sehingga perlu pengamanan dalam mengakses ruang server pada suatu perusahaan. Metode yang digunakan untuk menyelesaikan masalah yaitu metode *triangle face*. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang sitem kemanan dengan menggunakan *face recognition*. Kesimpulan dari penelitian tersebut adalah Pengenalan wajah menggunakan *metode triangle face* ini memiliki tingkat keakuratan 75%, kesalahan posistif 25% dan kesalahan negatif 0% sehingga dapat disimpulkan bahwa sistem ini cukup aman untuk diaplikasikan dalam system keamanan pintu ruang server.

Jurnal yang berjudul *Implementasi Push Notifikasion* Berbasis *Android* Untuk Sistem Monitoring Keamanan Rumah. (Siddik, 2020) meneliti monitoring terhadap keadaan rumah dilakukan dengan memantau pintu dan jendela rumah dengan menggunakan *sensor magnetic switch*, nantinya setiap sensor yang terdapat pada pintu dan jendela akan terhubung dengan *Raspberry Pi* yang tersingkronisasi dengan *firebase*. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang sitem kemanan rumah. Kesimpulan dalam penelitian tersebut yaitu implementasi *push notification* sebagai monitoring keamanan rumah dengan mengirimkan pesan *notification* yang dikirim

oleh *firebase* yang sudah tersingkronisasi dengan *raspberry pi* dan *sensor magnetic switch* dapat terkirim kepada *smartphone android* pemilik rumah.

Jurnal yang berjudul Perancangan *Smart Security Camera* Dengan *Model Image Processing* Menggunakan *Raspberry Pi.* (Candra et al., 2019) meneliti tentang rancangan sebuah kamera menjadi lebih cerdas dengan menggunakan mode *image processing* menggunakan perangkat *raspberry pi,* sehingga bisa memantau dan mengontrol suatu lokasi atau tempat dan memberikan laporan peringan melalui aplikasi *media social whatshap* ke smartphone yang sudah diseting ke perangkatnya. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang sitem kemanan rumah dengan memanfaaatkan *instan messenger* sebagai notifikasi. Kesimpulan dari penelitain tersebut adalah Sistem ini dapat diakses dengan menggunakan internet maupun tanpa harus terhubung dengan jaringan yang sama dengan Raspberry Pi. Camera Pi diatur di area yang ingin di pantau, video yang mengandung gerakan yang terdeteksi dan video langsung dapat diakses dari mana saja hanya dengan memasukkan IP statis yang ditetapkan ke Sistem di browser web.

Jurnal yang berjudul Sistem Pengenalan Wajah dengan *Algoritma Haar Cascade dan Local Binary Pattern Histogram* (Al-Aidid & Pamungkas, 2018) meneliti tentang penggunaan *algoritma haar cascade* sebagai pendeteksi wajah, maksud dari penelitian tersebut adalah agar sistem dapat mengenali wajah dari objek bukan wajah dengan jarak optimal antara 50-150 cm. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang *face recognition*. Kesimpulan dalam penelitian tersebut yaitu

metoda haar cascade digabungkan dengan *metoda local binary pattern histogram* dapat digunakan untuk mendetektsi dan mengenali wajah manusia, meskipun di sekitar dari manusia terdapat beberapa objek lain. Kecepatan yang dihasilkan cukup cepat, ini menunjukkan bahwa komputasi dari sistem ini cukup efektif.

Jurnal yang berjudul Pengembangan *Face Recognition* pada *Raspberry Pi* untuk Peningkatan Keamanan *Smart Home System.* (Gunawan et al., 2017) meneliti tentang sistem rumah pintar, maksud dari penelitian tersebut adalah untuk mengimplementasikan *ekstraksi* fitur dan *pengklasifikasi* dengan menggunakan *eigenface* dan PCA, sedangkan tujuan yang ingin dicapai adalah untuk meningkatkan sistem keamanan rumah, yang dapat dengan mudah mengunci dan membuka kunci pintu atau gerbang. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang sitem kemanan rumah. Kesimpulan dalam penelitian tersebut yaitu tingkat pengenalan ditemukan sekitar 90% ketika diuji dengan tiga orang. Sistem yang diusulkan ini dapat dihubungkan menggunakan internet ke sistem rumah pintar untuk kemampuan keamanan tambahan.

Jurnal yang berjudul Pengenalan Wajah Untuk Sistem Kehadiran Menggunakan *Metode Eigenface* dan *Euclidean Distance* (Syuhada et al., 2018) meneliti tentang pengenalan wajah untuk sistem kehadiran, maksud dari penelitian tersebut adalah bagaimana komputer yang sudah terkoneksi kamera webcam dapat mengenali wajah seseorang walaupun orang tersebut tidak secara langsung melakukan proses absensi. *Metode* yang digunakan untuk menyelesaikan masalah yaitu *metode eigenface* dan *euclidean distance*. Pembaruan dari penelitian tersebut

yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang *face recognition*. Kesimpulan dari penelitain tersebut adalah dengan Tingkat akurasi tertinggi yaitu sebesar 84 %. Preprocessing sangat mempengaruhi tingkat akurasi sistem dengan selisih 20 %. *Preprocessing* dengan normalisasi *histogram equalization* meningkatkan akurasi sebesar 70%. *Metode Eigenface* dan *Euclidean Distance* baik digunakan untuk melakukan pengenalan wajah jika banyak variasi data yang dimiliki degan kata lain semakin banyak variasi data yang dimiliki semakin akurat hasil pengenalannya.

Jurnal yang berjudul Implementasi *Face Detection* Dan *Recognition* Menggunakan *Python* Dengan *Numpy* Dan *Opencv* Menggunakan *Metode Haar-Cascade* Dan *Lbph (Local Binary Pattern Histogram)* (Face et al., 2021) meneliti tentang *face detection* dan *recognition*, maksud dari penelitian tersebut adalah untuk mendeteksi bagian wajah dari seseorang, wajah tersebut bisa didapatkan dari gambar maupun video. Metode yang digunakan untuk menyelesaikan masalah yaitu menerapkan *metode haar-cascde* untuk membangun sistem deteksi wajah dengan bahasa pemrograman *python* kemudian juga menggunakan metode *LBPH (Local Binary Paterrn Histogram)*. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang *face recognition.* Kesimpulan dari penelitian tersebut adalah Proses pengenalan wajah tidak optimal terhadap perubahan cahaya, jarak, atribut wajah, dan perubahan yang telalu ekstrim. Metode Haar-Cascade proses pendeteksian wajah dilakukan dengan mengklasifikasikan sebuah gambar setelah sebelumnya sebuah pengklasifikasi

dibentuk dari dataset sistem mampu mendeteksi wajah dan pengenalan wajah dapat mendeteksi objek wajah secara real time.

Jurnal yang berjudul *Automatic Email Alert on the Internet of Things-based Smart Motion Detection System* (Widiyasono et al., 2020) meneliti tentang sistem keamanan rumah berbasis *IoT* yang memanfaatkan sensor PIR, maksud dari penelitian tersebut adalah untuk mendeteksi gerak manusia yang terdeteski oleh sensor PIR maka secara otomatis mengirim peringatan email dengan melampirkan gambar. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang keamanan rumah. Kesimpulan dari penelitian tersebut adalah Sistem Deteksi Gerakan berbasis *Internet of Things* telah berhasil diimplementasikan dengan mekanisme pengiriman email alert secara otomatis dengan menambahkan hasil menembak ketika ada gerakan manusia, pada jarak antara 0-5 meter dengan berbagai kondisi cahaya, durasi rata-rata pengiriman email adalah 15 detik tergantung pada kualitas gambar ditangkap oleh Pi Kamera.

Jurnal yang berjudul Pengembangan Sistem Peringatan Otomatis Berbasis Internet Of Things Untuk Keamanan Rumah Berdasarkan Deteksi Pengenalan Objek (Bagus Sanjaya Putra & Widiyasono, 2021) meneliti tentang sistem keamanan rumah berbasis IoT, maksud dari penelitian tersebut adalah untuk mendeteksi gerak manusia yang terdeteski oleh *PI Camera* maka secara otomatis mengirim peringatan notifikasi ke telegram dengan melampirkan gambar. Pembaruan dari penelitian tersebut yang dianggap relevan dengan penelitian yang dilakukan yaitu meneliti tentang keamanan rumah. Kesimpulan dari penelitian tersebut adalah Berdasarkan hasil percobaan, sistem yang telah dibuat mampu

mendeteksi 13 object berbeda, pengujian Pi Camera berdasarkan jarak dapat terdeteksi sejauh 2 meter, dengan rata rata keberhasilan 95% pada jarak 1 meter dan 88% keberhasilan pada jarak 2 meter.

2.2.1. Matriks Penelitian

Tabel 2.1 menjelaskan tentang spesifik pembeda penelitian terdahulu dengan sekarang, ruang lingkup penelitian yang menjadi pembeda diantaranya modul dan platform yang dipakai, notifikasi yang dihasilkan, metode dan penelitian terdahulu termasuk ke dalam pengupayaan meningkatkan keamanan rumah.

Tabel 2.1 Matriks Penelitian

No	Peneliti/ Tahun	Judul	Ruang Lingkup Penelitian													
			Modul		Notification				Platform		Metode				Face Recognition	Keamanan Rumah
			Kamera	Sensor *PIR*	Telegram	SMS	*Email*	*Whatsapp*	Computer	Raspberry Pi	Eigenface	Support Vector Machine	CNN	Haar-Cascde		
1	Rudi Kurniawan, Antoni Zulius [2019]	*Smart Home Security menggunakan Face Recognition dengan Metode Eigenface berbasis Raspberry Pi*	v	v		V				v	v				v	v
2	Muhamad Irfan Kurniawan, Unang Sunarya, Rohmat Tulloh [2018]	*Internet of Things: Sistem Keamanan Rumah berbasis Raspberry Pi dan Telegram Messenger*	v	v	v					v						v
3	Dodon Yendri, Rahmi Eka Putri [2018]	*Sistem Pengontrolan dan Keamanan Rumah Pintar (Smart Home) berbasis Android*	v	v												v
4	Yuliza [2018]	*Detektor Keamanan Rumah melalui Telegram Messenger*		v	v					v						v

No	Peneliti/ Tahun	Judul	Ruang Lingkup Penelitian													
			Modul		Notification				Platform		Metode				Face Recognition	Keamanan Rumah
			Kamera	Sensor *PIR*	Telegram	SMS	*Email*	*Whatsapp*	Computer	Raspberry Pi	Eigenface	Support Vector Machine	CNN	Haar-Cascde		
5	Fadilah Eka Prasetiyo, Didik Setiyadi [2021]	*Sistem Pendeteksi Ancaman Keamanan Rumah Dengan menggunakan Telegram berbasis Internet Of Things*	v	v	v					v						v
6	Rajes Khana, Uus Usnul [2018]	*Rancang Bangun Sistem Keamanan Rumah berbasis Internet Of Things Dengan Platform Android*		v			v									v
7	M. Syarif Hidayatulloh, A.Yudi Permana, Wahyu Hadi Kristanto [2018]	*Pengenalan Wajah dengan Algoritma Support Vector Machine dan Sobel Edge Detection Berbasis Computer Vision dan Caffe Framework*	v									v			v	v

No	Peneliti/ Tahun	Judul	Ruang Lingkup Penelitian													
			Modul		Notification				Platform		Metode				Face Recognition	Keamanan Rumah
			Kamera	Sensor *PIR*	Telegram	SMS	*Email*	*Whatsapp*	Computer	Raspberry Pi	Eigenface	Support Vector Machine	CNN	Haar-Cascde		
8	Musakkarul Mu'minim Lambacing, Ferdiansyah [2020]	*Rancang Bangun New Normal Covid-19 Masker Detektor Dengan Notifikasi Telegram Berbasis Internet Of Things*	v	v	v					v						
9	Dedi Setiawan, Joni Eka Candra, Cosmas Eko Suharyanto [2019]	*Perancangan Sistem Pengontrol Keamanan Rumah Dengan Smart CCTV Menggunakan Arduino Berbasis Telegram*	v	v	v											v
10	Fadli Sirait [2015]	*Sistem Monitoring Keamanan Gedung berbasis Rasberry Pi*	v	v				v		v						

No	Peneliti/ Tahun	Judul	Ruang Lingkup Penelitian													
			Modul		Notification				Platform		Metode					
			Kamera	Sensor *PIR*	Telegram	SMS	*Email*	*Whatsapp*	Computer	Raspberry Pi	Eigenface	Support Vector Machine	CNN	Haar-Cascde	Face Recognition	Keamanan Rumah
11	Indra Dharma Wijaya, Usman Nurhasan, Mula Agung Barata [2017]	*Implementasi Raspberry Pi Untuk Rancang Bangun Sistem Keamanan Pintu Ruang Server Dengan Pengenalan Wajah Menggunakan Metode Triangle Face*	v							v					v	
12	Mohd. Siddik [2020]	*Implementasi Push Notifikasion Berbasis Android Untuk Sistem Monitoring Keamanan Rumah*	v	v		V				v						v
13	Hardisal, Rudi Arif Candra, Ihsan, Dirja Nur Ilham, Erwinsyah	*Perancangan Smart Security Camera Dengan Model Image Processing*	v					v		v						v

	Sipahutar [2019]	*Menggunakan Raspberry Pi*														
14	Sayeed Al-Aidid dan Daniel S. Pamungkas [2018]	*Sistem Pengenalan Wajah dengan Algoritma Haar Cascade dan Local Binary Pattern Histogram*	v											v	v	
No	**Peneliti/ Tahun**	**Judul**	**Ruang Lingkup Penelitian**													
			Modul		**Notification**				**Platform**		**Metode**				**Face Recognition**	**Keamanan Rumah**
			Kamera	**Sensor *PIR***	**Telegram**	**SMS**	***Email***	***Whatsapp***	**Computer**	**Raspberry Pi**	**Eigenface**	**Support Vector Machine**	**CNN**	**Haar-Cascde**		
15	Teddy Surya Gunawan, Muhammad Hamdan Hasan Gani, Farah Diyana Abdul Rahman, Mira Kartiwi [2017]	*Development of Face Recognition on Raspberry Pi for Security Enhancement of Smart Home System*	v							v	v				v	v
16	Fahmi Syuhada, I Gede Pasek Suta Wijaya, Fitri Bimantoro [2018]	*Pengenalan Wajah Untuk Sistem Kehadiran Menggunakan Metode Eigenface dan Euclidean Distance*	v								v				v	
17	Majid Al-Kuwari, Abdulrhman Ramadan, Yousef Ismael,	*Implementasi Face Detection Dan Recognition menggunakan Python dengan*	v											v	v	

	Laith Al-Sughair, Adel Gastli, [2018]	*Numpy dan Opencv menggunakan Metode Haar-cascade dan Lbph (local binary pattern histogram)*														

No	Peneliti/ Tahun	Judul	Ruang Lingkup Penelitian													
			Modul		Notification				Platform		Metode				Face Recognition	Keamanan Rumah
			Kamera	Sensor *PIR*	Telegram	SMS	*Email*	*Whatsapp*	Computer	Raspberry Pi	Eigenface	Support Vector Machine	CNN	Haar-Cascde		
18	Helmi Firmansah [2019]	*Automatic Email Alert On The Internet Of Things-Based Smart Motion Detection System*	v	v			v			v						v
19	Bagus Sanjaya Putra [2021]	*Pengembangan Sistem Peringatan Otomatis Berbasis Internet Of ThingsUntuk Keamanan Rumah Berdasarkan Deteksi Pengenalan Objek*	v		v					v						v
20	Dede Yusup [2022]	*Pengembangan Sistem Peringatan Otomatis Berbasis Internet Of ThingsUntuk*	v		v				v				v		v	v

		Keamanan Rumah Berdasarkan Pengenalan Wajah														

2.2. Tinjauan Pustaka

2.2.1. Sistem Keamanan Rumah Berbasis *IoT*

Sistem keamanan rumah berbasis *IoT* dapat didefinisikan sebagai pemantauan sebagian rumah maupun rumah secara keseluruhan dari lokasi yang letaknya jauh dan terpusat (Tanwar et al., 2017), dan juga memungkinkan untuk melakukan kontrol dan monitoring keadaan rumah menggunakan sistem keamanan rumah dari jarak jauh secara *real time* melalui jaringan *internet* yang bisa kita akses kapan dan dimana saja (Khana & Uus Usnul, 2018).

Internet of Things (IoT) adalah adalah pengembangan internet baru yang menghubungkan miliaran sensor dan perangkat lain ke internet. Perangkat *IoT* dapat berkomunikasi secara langsung, dan mengijinkan pengguna internet untuk mengakses, menyalurkan informasi dan memanfaatkan data yang diberikan (Yavari et al., 2017).

2.2.2. Face Recognition

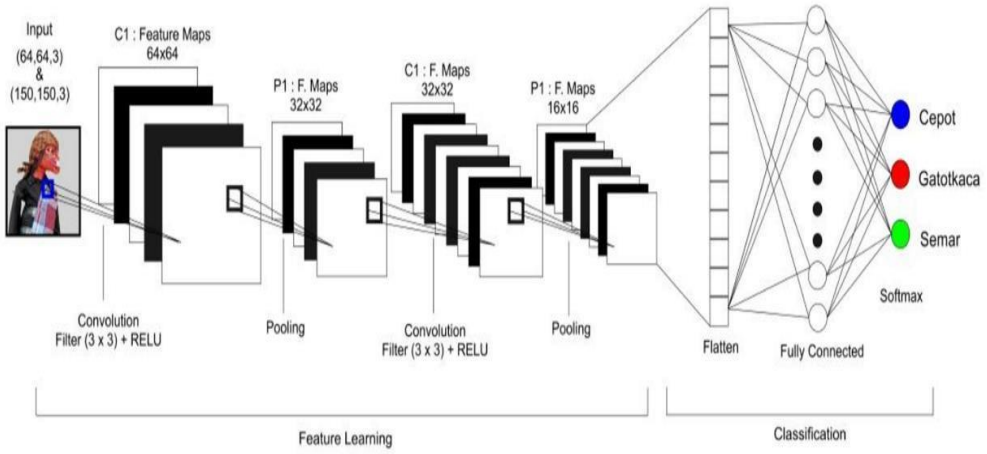
Face Recognition merupakan salah satu teknologi biometrik yang telah banyak diimplementasikan dalam sistem *security* untuk membantu dalam mengidentifikasi wajah seseorang (Rudi Kurniawan & Zulius, 2019). Pengenalan wajah bekerja dengan mendeteksi citra wajah dari objek yang diinputkan kemudian

dilakukan klasifkasi terhadap citra yang tersimpan didalam basis data (Hidayatulloh et al., 2020).

2.2.3. Convolutional Neural Network

CNN *(Convolutional Neural Network*) adalah salah satu jenis *deep learning* yang paling popular di berbagai bidang terkait dengan pengenalan pola dari pemrosesan gambar, identifikasi objek hingga pengenalan wajah. Bagian paling bermanfaat dari CNN adalah mengurangi batasan dalam ANN *(Artificial Neural Network*). Pencapaian ini telah mendorong para peneliti untuk beralih ke model yang lebih besar dalam menangani tugas-tugas kompleks, yang tidak mungkin dilakukan dengan jaringan saraf tiruan biasa (Albawi, dkk 2017:1-6).

Secara umum tipe layer CNN terdiri dari *feature learning/extraction* dan *classification layer*. Dimana *feature extraction* terdiri dari *convolution layer*, *activation* dan *pooling layer*. Sementara *classification layer* terdiri dari *flatten*, *fully connected layer* dan *output layer*. CNN memiliki kinerja yang sangat baik dalam mengatasi masalah pembelajaran mesin (*machine learning*) dan *deep learning*. Khususnya aplikasi yang berhubungan dengan data gambar, seperti kumpulan data klasifikasi gambar terbesar *(Image Net*), computer vision dan*natural languan*ge *processing*. Arsitektur CNN dapat dikatakan mirip dengan polahubungan neuron atau sel saraf pada otak manusia. CNN terinspirasi dari visual cortex, yaitu bagian pada otak yang bertugas untuk memproses informasi dalam bentuk visual (Albawi, dkk 2017:1-6). Arsitektur dari CNN *(Convolution Neural Network*) ditunjuk kan pada Gambar 2.1.

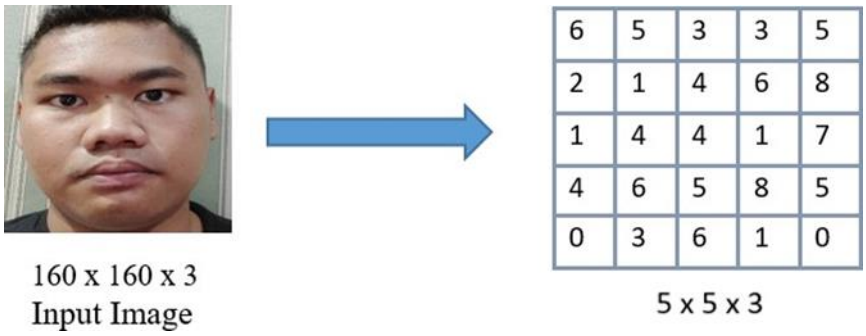


Gambar 2.1 Contoh Arsitektur *Convolutional Neural Network*

a. *Convolution Layer*

Lapisan konvolusi adalah lapisan untuk proses manipulasi gambar agar menghasilkan gambar baru. Lapisan-lapisan pada CNN bekerja secara hierarki, artinya output dari lapisan pertama akan digunakan sebagai input pada lapisan selanjutnya. Terdapat banyak parameter yang dapat diubah untuk melakukan modifikasi sifat tiap lapisan, yaitu *kernel*, *stride* dan *padding* dan lain-lain. *Kernel* dengan nama lain (*feature detector*/*filter*) dilakukan untuk mengekstraksi objek dari gambar sebagai data masukan. *Kernel* tersebut mendeteksi karakter dari gambar tersebut seperti mulut, hidung dan mata. *Kernel* dilakukan secara berulang hingga menghasilkan data gambar dengan kualitas lebih baik, menghilangkan derau (*noise*) serta menghaluskan gambar. *Stride* berfungsi sebagai pengontrol *kernel* yang diterapkan pada data masukan dan sebagai penentu jumlah pergeseran *filter*. Misal, jika nilai *stride* adalah 1, maka *kernel* pada *layer* konvolusi akan bergeser sebanyak 1 *piksel* secara *horizontal* lalu *vertikal*. Parameter *padding* adalah penambahan ukuran piksel dengan nilai tertentu disekitar data *input*, hal ini dilakukan karena

proses pada *kernel* mereduksi ukuran dari matriks gambar aslinya sehingga diperlukan *padding* untuk tetap mempertahankan ukuran matriks sebenarnya (Jurjawi, 2020). Berikut ini adalah pembahasan untuk proses lapisan konvolusi.



Gambar 2.2 Sampel Gambar Data Masukan

Gambar 2.2 merupakan gambar input *image* yang mewakili *dataset*. Karena *input image* memiliki ukuran piksel 160 x 160, maka peneliti hanya mengambil sebagian piksel saja atau bisa disebut sebagai matriks karena memiliki baris dan kolom yang akan di jadikan sampel dalam proses konvolusi. Setiap piksel atau matriks memiliki nilai, nilai ini disebut sebagai nilai piksel. Dimana nilai piksel itu memiliki rentang 0 - 255, dan sampel nilainya dapat dilihat pada matriks yang berukuran 5 x 5 x 3 pada Gambar 2.2. Proses penyederhanaan ukuran piksel/matriks tersebut dilakukan agar lebih mudah dalam memahami proses konvolusi pada *convolution layer*, dan perhitungannya bisa lebih sederhana. Angka 3 diatas maksudnya adalah bahwa gambar tersebut memiliki 3 *channel* warna yaitu *red*, *green*, *blue* atau biasa disebut RGB, dan pada penelitian ini peneliti menggunakan gambar RGB sebagai dataset.

Hasil akhir dari proses konvolusi menggunakan rumus seperti pada Persamaan 2.1 (Suyanto dan Mandala, 2019).

$$Output = \frac{W-F+2P}{S} + 1 \quad \text{.....2.1}$$

Keterangan:

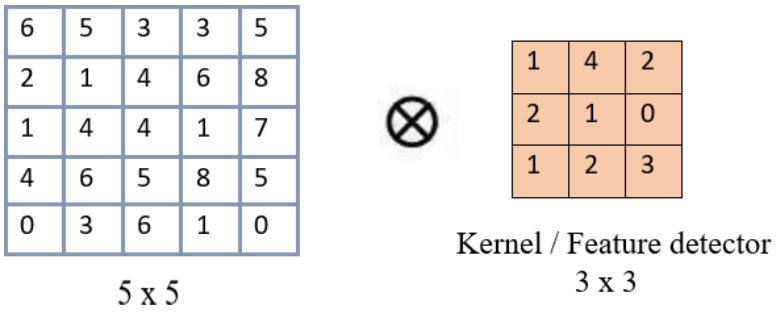
W = ukuran gambar

F = ukuran kernel/filter

P = *padding*

S = *stride*

Proses konvolusi dengan kernel dapat dilihat pada Gambar 2.3.



Gambar 2.3 Proses konvolusi dengan kernel/feature detector 3x3

Diketahui pada Gambar 2.3 bahwa:

W = 5x5

F = 3x3

P = 0

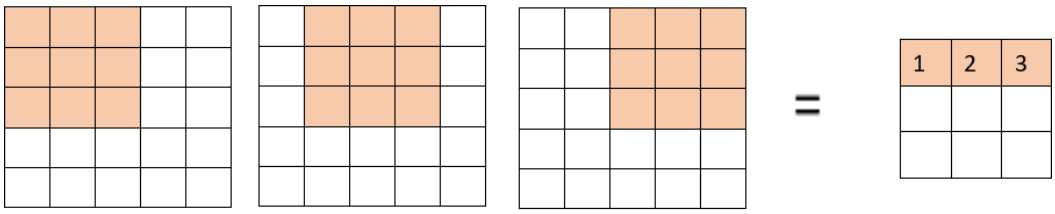
S = 1

Sehingga hasil perhitungannya sebagai berikut:

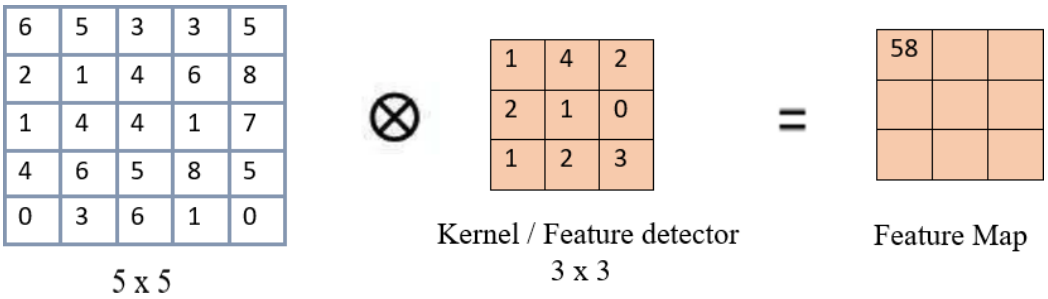
$$Output = \frac{5 - 3 + 2(0)}{1} + 1$$

$$Output = 3$$

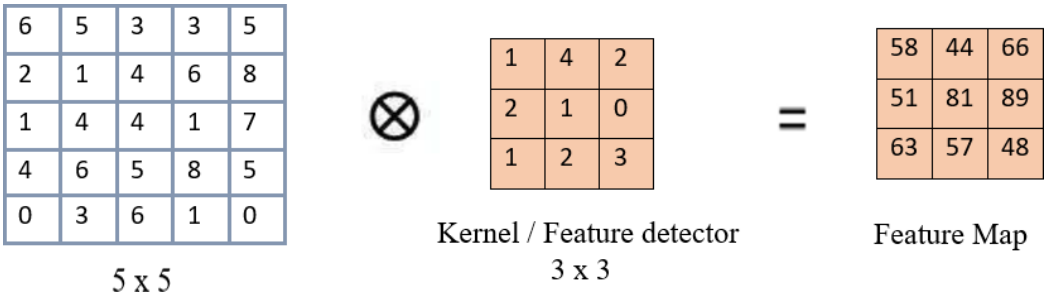
Dari perhitungan di atas maka didapatkan *output* (*feature map*) dengan ukuran 3x3.



Gambar 2.4 Sampel pergeseran dengan nilai *stride* = 1 dan *kernel* 3x3



Gambar 2.5 Proses konvolusi pergeseran pertama



Gambar 2.6 Proses konvolusi pergeseran terakhir

Proses konvolusi untuk setiap pergeseran pada sampel Gambar 2.5 dan Gambar 2.6 dilakukan melalui perkalian dot product. Perhitungannya adalah sebagai berikut:

Posisi 1 :

(6x1) + (5x4) + (3x2) + (2x2) + (1x1) + (4x0) + (1x1) + (4x2) + (4x3) = 58

Posisi 2 :

(5x1) + (3x4) + (3x2) + (1x2) + (4x1) + (6x0) + (4x1) + (4x2) + (1x3) = 43

Posisi 3 :

(3x1) + (3x4) + (5x2) + (4x2) + (6x1) + (8x0) + (4x1) + (1x2) + (7x3) = 66

Posisi 4 :

(2x1) + (1x4) + (4x2) + (1x2) + (4x1) + (4x0) + (4x1) + (6x2) + (5x3) = 51

Posisi 5 :

(1x1) + (4x4) + (6x2) + (4x2) + (4x1) + (1x0) + (6x1) + (5x2) + (8x3) = 81

Posisi 6 :

(4x1) + (6x4) + (8x2) + (4x2) + (1x1) + (7x0) + (5x1) + (8x2) + (5x3) = 89

Posisi 7 :

(1x1) + (4x4) + (4x2) + (4x2) + (6x1) + (5x0) + (0x1) + (3x2) + (6x3) = 63

Posisi 8 :

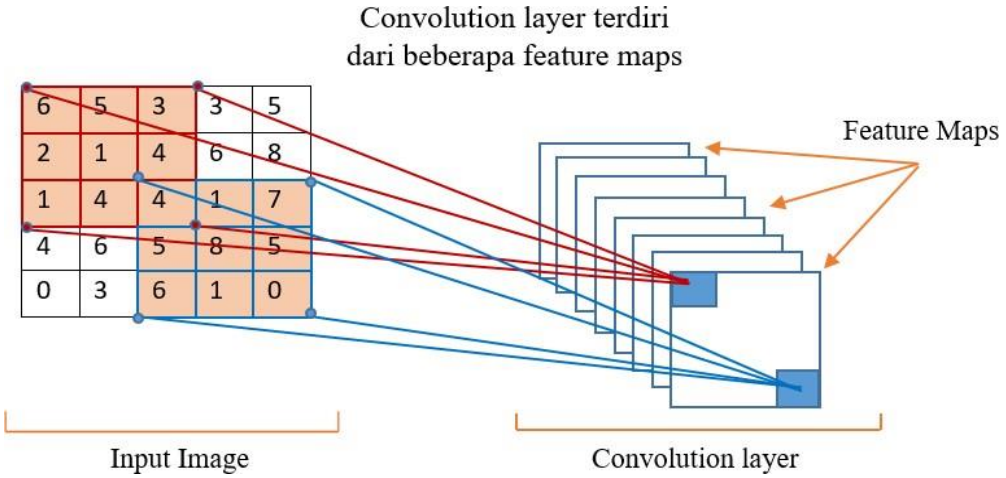
(4x1) + (4x4) + (1x2) + (6x2) + (5x1) + (8x0) + (3x1) + (6x2) + (1x3) = 57

Posisi 9 :

(4x1) + (1x4) + (7x2) + (5x2) + (8x1) + (5x0) + (6x1) + (1x2) + (0x3) = 48

Dapat dilihat bahwa gambar awal yang berukuran 5×5 berkurang menjadi lebih kecil yaitu 3×3. Ini memang salah satu tujuan utama *kernel/feature detector*. Bisa dibayangkan jika mengolah gambar yang berukuran besar dengan resolusi tinggi, maka bisa sampai ribuan pixel. Melalui *kernel/feature detector* proses pengolahan gambar menjadi semakin cepat karena data *pixel* yang diolah juga semakin kecil (sedikit).

Terdapat banyak jumlah *kernel*/*feature* detector dalam proses CNN, maka ada banyak *feature map*. Kumpulan dari *feature map* disebut dengan *convolution layer*. Ilustrasinya ditunjukkan pada Gambar 2.7.



Gambar 2.7 Proses pembuatan *convolution layer* dari beberapa *feature maps*

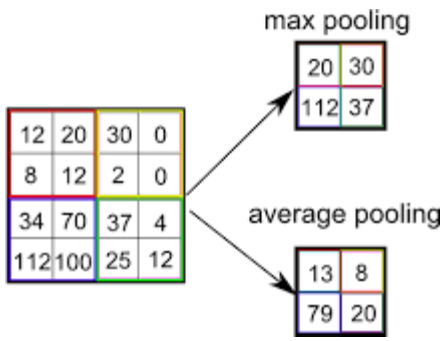
b. *Activation Function*

Sebelum memasuki ke proses *pooling*, maka harus menggunakan fungsi aktivasi, fungsi aktivasi bertujuan untuk menentukan apakah sebuah neuron harus aktif atau tidak berdasarkan dari *weighted sum* dari *input*. Fungsi aktivasi yang digunakan pada penelitian ini yaitu ReLU (*Rectifier Linear Unit*), karena ReLU memiliki rentang nilai dari 0 sampai tak hingga. Nilai yang ada pada hasil konvolusi

yang bernilai *negative* akan diubah dengan aktivasi ini menjadi nol, sehingga dengan menggunakan ReLU, hasil akhir dari matriks yang didapatkan adalah tetap karena nilai pada elemen matriks tidak ada yang bernilai kurang dari nol (Jurjawi, 2020).

c. *Pooling Layer*

Pooling layer merupakan salah satu teknik untuk melakukan proses pencarian fitur yang sudah didapat di tahap sebelumnya (Convolution). Pooling bekerja dengan mengurangi ukuran matriks menggunakan operasi pooling (penggabungan). Dari hasil convolution layer pada lapisan sebelumnya, teknik pooling ini akan mencari fitur dengan tetap mempertahankan fleksibilitas yang tinggi. Ia tidak peduli apakah gambarnya miring ke kanan, kiri, diputar, dibesarkan, dikecilkan, berbeda teksturnya, warnanya dan lain sebagainya. Ia hanya peduli dengan fitur yang ia cari. Banyak metode pooling yang bisa digunakan diantaranya yaitu MaxPooling, AveragePooling, SumPooling dan lain – lain (Jurjawi, 2020). Ilustrasi ditunjukkan pada Gambar 2.8.



Gambar 2.8 Contoh *Pooling* dengan teknik *MaxPooling* dan *AveragePooling*

Gambar 2.8 merupakan contoh teknik *MaxPooling* dan *AveragePooling* dengan *kernel* 2 x2 dan *stride* = 2. Untuk *MaxPooling*, pada setiap pergeseran *kernel*, nilai maximum pada area 2x2 pixel tersebut yang akan dipilih, sedangkan *AveragePooling* akan memilih nilai rata-ratanya. Rumus yang digunakan pada *pooling layer* ini adalah seperti pada Persamaan 2.2 (Suyanto dan Mandala, 2019).

$$Output = \frac{W-F}{S} + 1 \quad \text{.....2.2}$$

Keterangan:

W = 5x5

F = 3x3

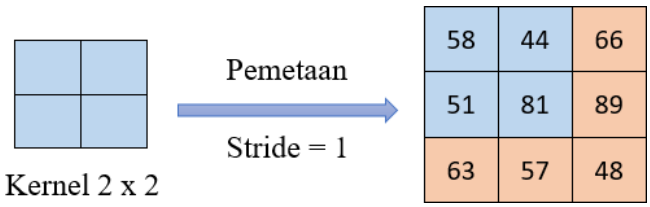
S = 1

Hasil akhir dari proses konvolusi akan digunakan kembali pada proses *pooling* atau dengan kata lain, sebagai masukan di lapisan *pooling*. Ilustrasinya ditunjukkan pada Gambar 2.9.

58	44	66
51	81	89
63	57	48

Gambar 2.9 Hasil akhir dari lapisan konvolusi sebagai *input* pada *pooling*

Proses pooling dengan *kernel*/*filter* ditunjukkan pada Gambar 2.10.



Gambar 2.10 Pooling dengan kernel 2x2

Diketahui pada Gambar 2.10 bahwa:

W = 3x3

F = 2x2

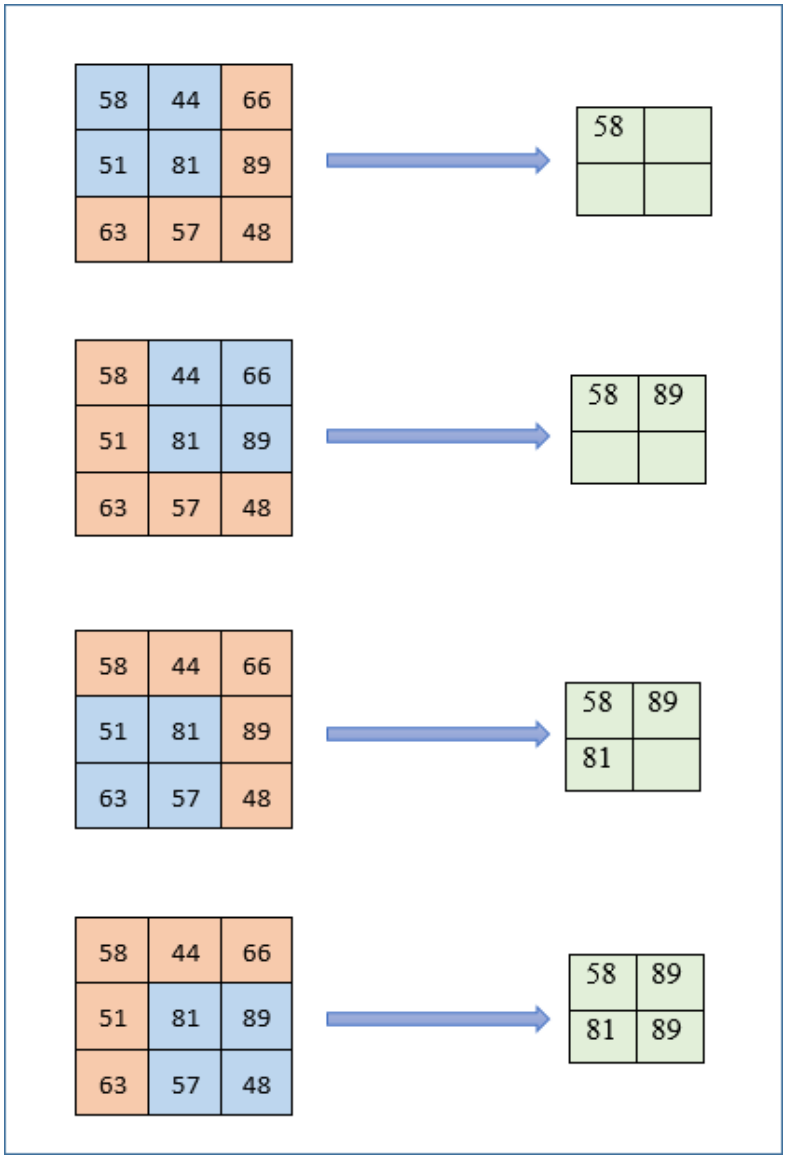
S = 1

Sehingga hasil perhitungannya sebagai adalah :

$$Output = \frac{3-2}{1} + 1$$

$$Output = 2$$

Dari perhitungan diatas maka didapatkan output (pooling) dengan ukuran 2x2. Proses pooling dengan metode MaxPooling ditunjukkan pada Gambar 2.11.



Gambar 2.11 Proses *pooling* dengan metode *MaxPooling*

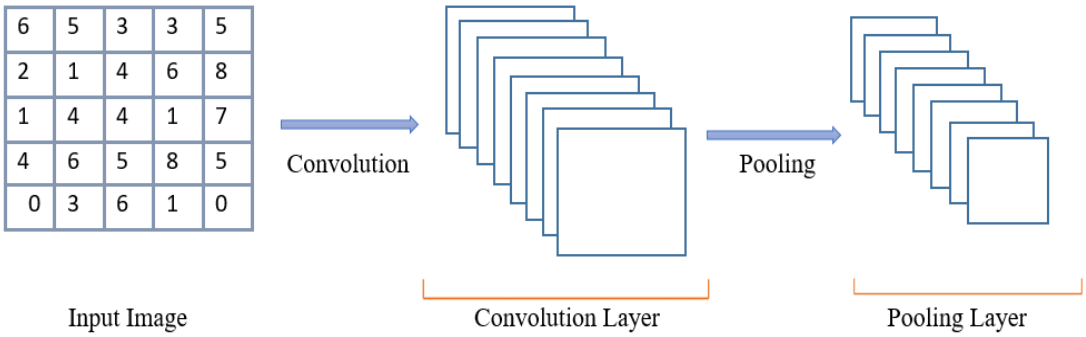
Berdasarkan Gambar 2.11 bahwa elemen yang diarsir dengan warna biru adalah matriks dengan *kernel* 2x2. Dari hasil kernel tersebut, fungsi *MaxPooling* mengambil nilai maksimal. Nilai maksimal tersebut disimpan dalam matriks baru. Setelah melakukan proses pengambilan nilai maksimal pada satu *kernel*, selanjutnya *kernel* 2x2 bergerak dengan *stride* = 1 untuk mengulang kembali fungsi *MaxPooling*, yaitu mengambil nilai tertinggi dari *kernel* 2x2 dan disimpan pada

matriks yang telah ditetapkan sebelumnya. Proses *pooling* berulang sampai *kernel* berada pada matriks paling akhir dari proses perpindahan *stride* (Jurjawi, 2020). Hasil akhir dari pooling layer adalah matriks 2 x 2 dengan elemen matriks yang dapat dilihat pada Gambar 2.12.

58	89
81	89

Gambar 2.12 Hasil akhir *pooling layer*

Dari proses *pooling* ini akan dihasilkan *pooling layer* (*pooled feature maps*). Tentunya dalam CNN kita melakukan beberapa kali proses *pooling* sehingga ada banyak *pooling layer* yang dihasilkan. Tampilannya seperti pada Gambar 2.13.

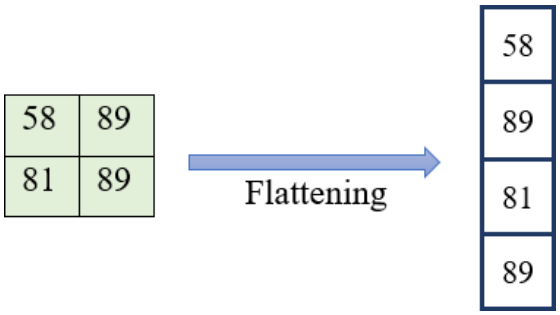


Gambar 2.13 Proses lapisan CNN dari awal hingga menjadi *pooling layer*

d. *Flatten Layer*

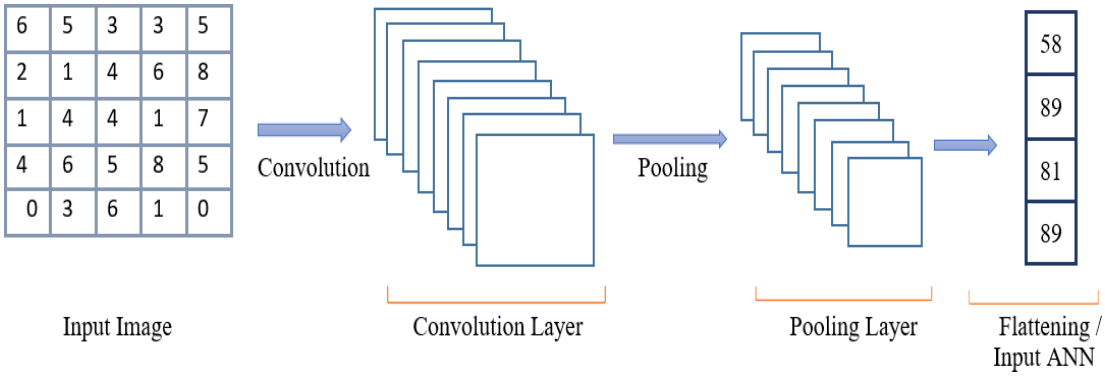
Flatten merupakan tahapan yang lebih sederhana jika dibanding dengan *convolution* dan *pooling layer*. Tahapan *flattening* bekerja dengan mengubah dari matriks yang ada di *pooling layer* menjadi satu kolom saja (sebuah vektor tunggal). Dimana vektor ini nantinya akan menjadi bagian dari *input layer* di lapisan *fully*

connected layer atau ANN *(Artificial Neural Network)* (Jurjawi, 2020). Ilustrasinya ditunjukkan pada Gambar 2.14.



Gambar 2.14 Proses *flattening*

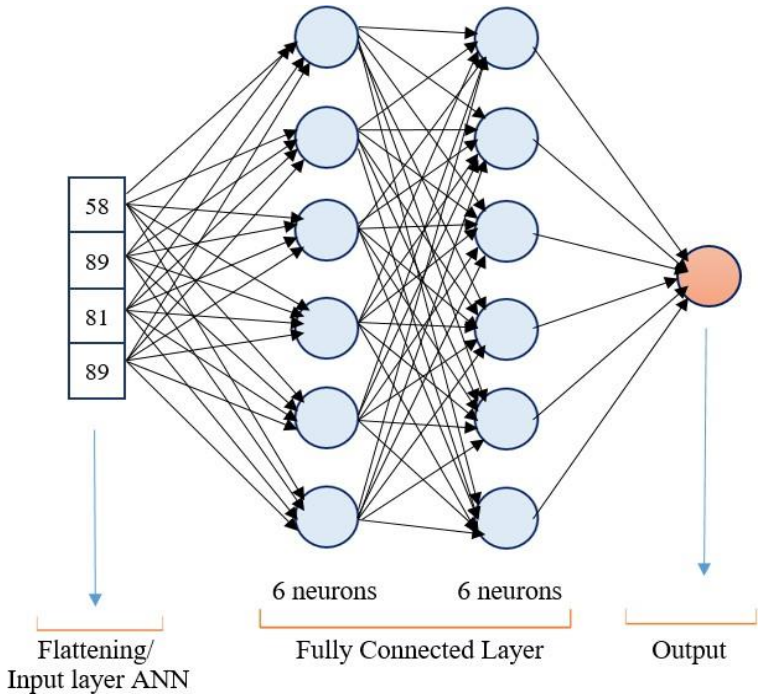
Setelah proses *flattening* dilakukan, maka hasilnya dimasukkan ke dalam input layer sebuah ANN. Semua hasil dari *pooling layer* (*pooled featute maps*) akan menjadi 1 vektor saja. Jadi, jika sebuah *pooling layer* berukuran 2×2 matriks, maka ia akan menjadi 1 vektor dengan 4 baris. Jika ada 8 *pooling layer* dengan ukuran yang sama (2×2), maka akan ada 1 vektor dengan 32 baris sebagai input untuk ANN (Jurjawi, 2020). Tampilan hingga proses *flattening* ditunjukkan pada Gambar 2.15.



Gambar 2.15 Proses lapisan CNN dari awal hingga menjadi *flattening*

e. *Fully Connected Layer*

Lapisan ini merupakan hasil dari proses flattening dimasukkan ke dalam struktur ANN (*Artificial Neural Networks*) yang utuh. Seperti biasa, ANN terdiri dari 3 bagian, yaitu *input layer*, *hidden layer*, dan *output layer*. Hal yang perlu diperhatikan adalah di dalam model *deep learning* ANN (artificial neural networks) bahwa setiap neuron (*nodes*) tidaklah harus saling terhubung dengan nodes di depannya. Dalam konteks CNN, maka semua nodes dari ANN harus terhubun dengan nodes di depan dan di belakangnya, oleh karena itu disebut dengan *fully connected* (Jurjawi, 2020). Ilustrasi dari *fully connected layer* ditunjukkan seperti pada Gambar 2.16.



Gambar 2.16 Lapisan *fully connected layer*

Gambar 2.16 merupakan lapisan *fully connected layer*, dimana nilai inputnya diperoleh dari hasil dari *flattening*. Lapisan *fully connected layer* disebut juga sebagai *hidden layer,* dimana banyaknya jumlah hidden layer tidak ada batasan ataupun bebas dalam menentukannya, dan contoh diatas memiliki 2 lapis *hidden layers*. Banyaknya jumlah *neurons / nodes / perceptron* pada setiap lapis dari *hidden layer* juga tidak ada batasan ataupun bebas dalam menentukannya, tetapi dalam hal ini harus disesuaikan dengan *hardware* komputer. Karena semakin banyak jumlah *hidden layer* dan neuronnya, maka proses komputasinya akan semakin lama dan berat. Pada contoh Gambar 2.16, diambil sampel sebanyak 6 *neurons* untuk setiap lapis dari *hidden layer*.

f. *Output Layer*

Output layer merupakan lapisan terakhir pada CNN maupun model *deep learning* lainnya. Pada lapisan ini banyak metode ataupun algortima yang bisa digunakan diantaranya *binary, sigmoid, softmax* dan lain-lain. Pada penelitian ini, peneliti menggunakan fungsi aktivasi *softmax* sebagai algoritma outputnya karena jumlah kelas yang lebih dari dua (*multi-class*). Fungsi *softmax* berfungsi untuk menghitung probabilitas pada semua label untuk semua kelas. Probabilitas tertinggi dari suatu label itu nantinya yang akan ditampilkan sebagai *output* pada saat melakukan prediksi (Jurjawi, 2020). Nilai *softmax* dapat dihitung menggunakan Persamaan 2.3.

$$f_j(z) = \frac{e^{zj}}{\sum_k e^{zk}} \quad \text{..... 2.3}$$

Notasi f_j menunjukkan hasil fungsi untuk setiap elemen ke-j pada vektor *output* kelas. Argumen z merupakan hipotesis yang diberikan olehmodel pelatihan atau bobot *(weight)* hasil pelatihan pada *hidden layer* sebelumnyasupaya dapat diklasifikasikan oleh fungsi *softmax*. Nilai dari *softmax* selalu bernilaisatu karena berisi probabilitas dari semua label (Jurjawi, 2020).

Diketahui *hidden layer* memiliki 6 *neurons* sesuai pada lapisan *fully connected layer*, dan anggap setiap *neuron* sudah memiliki bobot yang di simbolkan dengan z. Dimana:

$z_0 = 5$ $z_3 = 0,1$ $e = 2,718$
$z_1 = 0,5$ $z_4 = 1$
$z_2 = 0,2$ $z_5 = 0,8$

Maka nilai *softmax* untuk semua label yaitu:

$$= \frac{}{2{,}718^{5} + 2{,}718^{0{,}5} + 2{,}718^{0{,}2} + 2{,}718^{0{,}1} + 2{,}718^{1} + 2{,}718^{0{,}8}} +$$

$$= \frac{2{,}718^{0{,}5}}{2{,}718^{5} + 2{,}718^{0{,}5} + 2{,}718^{0{,}2} + 2{,}718^{0{,}1} + 2{,}718^{1} + 2{,}718^{0{,}8}} +$$

$$= \frac{2{,}718^{0{,}2}}{2{,}718^{5} + 2{,}718^{0{,}5} + 2{,}718^{0{,}2} + 2{,}718^{0{,}1} + 2{,}718^{1} + 2{,}718^{0{,}8}} +$$

$$= \frac{2{,}718^{0{,}1}}{2{,}718^{5} + 2{,}718^{0{,}5} + 2{,}718^{0{,}2} + 2{,}718^{0{,}1} + 2{,}718^{1} + 2{,}718^{0{,}8}} +$$

$$= \frac{2{,}718^{1}}{2{,}718^{5} + 2{,}718^{0{,}5} + 2{,}718^{0{,}2} + 2{,}718^{0{,}1} + 2{,}718^{1} + 2{,}718^{0{,}8}} +$$

$$= \frac{2{,}718^{0{,}8}}{2{,}718^{5} + 2{,}718^{0{,}5} + 2{,}718^{0{,}2} + 2{,}718^{0{,}1} + 2{,}718^{1} + 2{,}718^{0{,}8}}$$

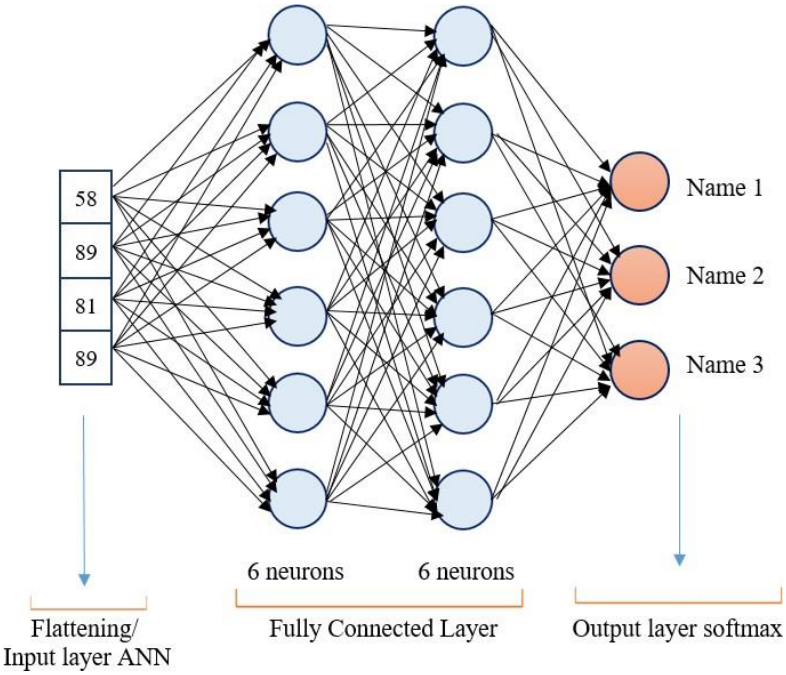
$$= \frac{49{,}0105 + 1{,}4747 + 1{,}2213 + 1{,}0809 + 2{,}178 + 1{,}8640}{2{,}718^{0{,}5} + 2{,}718^{0{,}5} + 2{,}718^{0{,}2} + 2{,}718^{0{,}1} + 2{,}718^{1} + 2{,}718^{0{,}8}}$$

$$= \frac{56{,}8294}{49{,}0105 + 1{,}4747 + 1{,}2213 + 1{,}0809 + 2{,}178 + 1{,}8640}$$

$$= \frac{56{,}8294}{56{,}8294}$$

$$= \mathbf{1}$$

Lapisan *output* dengan aktivasi fungsi *softmax* terlihat seperti Gambar 2.17.



Gambar 2.17 Lapisan *output* dengan aktivasi *softmax*

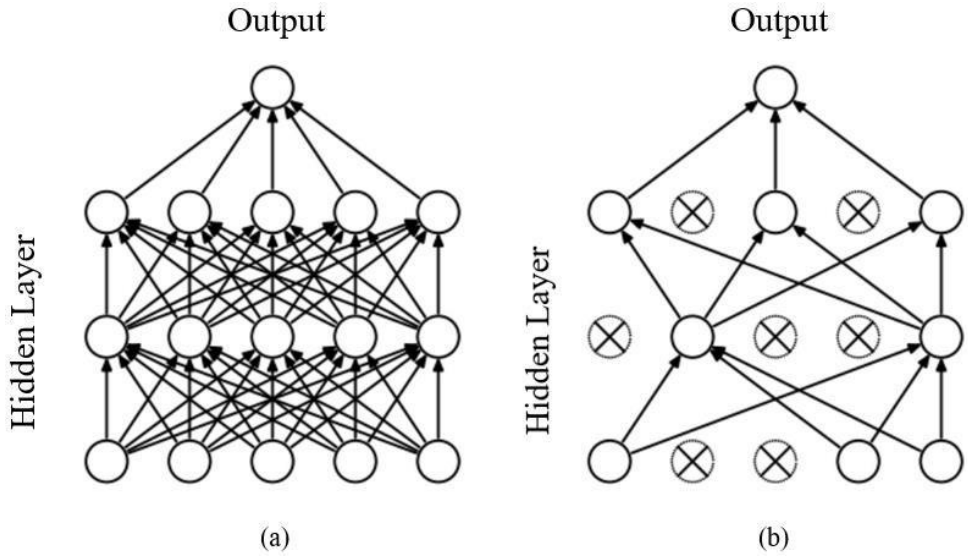
Jumlah *output* pada Gambar 2.17 berjumlah tiga kelas, dimana jumlah ini hanyalah sampel dari jumlah aslinya yang berjumlah 30 kelas. Jumlah kelas pada *output layer* tidak memiliki ketentuan dan harus disesuaikan dengan kebutuhan.

g. *Dropout Regularization*

Dropout merupakan teknik regularisasi jaringan syaraf *(neural network)* dimana beberapa akan dipilih secara random dan tidak dipakai (diputus) selama training. *Neuron - neuron* ini dibuang juga secara *random*. Hal ini berarti bahwa kontribusi *neuron* yang dibuang akan dihentikan sementara dan bobot baru juga tidak diterapkan pada *neuron*. *Rate* dari *dropout* berkisar antara 0 hingga 1. Tujuandari digunakannya *dropout* adalah untuk mengurangi

overfitting, dimana *overfitting*yaitu kondisi ketidakseimbangan nilai *loss* dan *accuracy* antara *training* dan *validation*, yang berarti jarak antara *training loss* dan *validation loss* serta *trainingaccuracy* dan *validation accuracy* memiliki *gap* yang cukup jauh, dan kondisi *overfitting* bisa menyebabkan salah prediksi pada saat pengujian dengan data asli.

Pengunaan *dropout* dapat dilakukan pada setiap *layer* baik di *convolution layer* maupun *hidden layer*, tetapi pengunaan *dropout* juga memiliki kekurangan yaitu dapat menghilangkan informasi pada lapisan-lapisan sebelumnya, dan bisa jadi *neurons* yang diputus berisi informasi hasil komputasi yang penting dan dibutuhkan untuk klasifikasi. Maka dari itu, harus banyak melakukan eksperimen terhadap pengunaan *dropout* agar menghasilkan *traning model* yang bagus dan dapat mengurangi *overfitting*, karena kondisi *overfitting* sangat umum terjadi pada *machine learning* dan *deep learning* (Suyanto dan Mandala, 2019). Ilustrasi *dropout* ditunjukkan pada Gambar 2.18.



Gambar 2.18 (a) *Hidden layer* sebelum menggunakan *dropout*

(b) *Hidden layer* setelah menggunakan *dropout*

2.2.4. OpenCV

OpenCV (*Open Source Computer Vision Library*) adalah pustaka perangkat lunak *machine learning*. *OpenCV* dibangun untuk menyediakan infrastruktur umum untuk *computer vision applications* dan untuk mempercepat penggunaan *machine perception* dalam produk komersial, menjadi produk berlisensi *BSD*, *OpenCV* memudahkan *enterprise* untuk menggunakan dan memodifikasi kodenya (Bagus Sanjaya Putra & Widiyasono, 2021).

2.2.5. WebCam

Webcam merupakan singkatan dari *Web* dan *Camera* atau juga disebut kamera *real-time* yang gambarnya dapat di lihat secara *online* melaului internet. Webcam adalah kamera video digital kecil yang dihubungkan ke komputer melalui port USB. Dasarnya webcam tidak terdapat penyimpanan data dan hasil perekaman yang didapat langsung ditransfer ke komputer (Yendri & Putri, 2018).

Gambar 2.19 *Webcam*

2.2.6. Telegram Messenger

Telegram adalah aplikasi pesan singkat yang diluncurkan pada tahun 2013 untuk banyak platform, di antaranya *android, IOS, windows, mac OS,* dan *linux*. Telegram merupakan aplikasi pesan singkat yang memungkinkan kita untuk bertukar pesan tanpa biaya *SMS*, karena telegram menggunakan internet. Selain bisa mengirim data dengan limit yang besar, aplikasi telegram juga lebih ringan dibandingkan dengan aplikasi pesan singkat lainya. Terdapat juga fitur *Bot* yang dilengkapi dengan AI (*Artificial Intellegence* atau kecerdasan buatan). *Telegram* memiliki fitur *secret chat* sehingga lebih aman dari pada *Whatsapp* (Yuliza, 2018).