

Pengenalan Ruang Kosong pada Tempat Parkir Mobil Menggunakan Convolutional Neural Network (CNN)

Rezky Riadhi Shalhan¹, Ema Rachmawati²

^{1,2}Fakultas Informatika, Universitas Telkom, Bandung

¹riadhishalhan@students.telkomuniversity.ac.id, ²emarachmawati@telkomuniversity.ac.id

Abstrak

Parkir di tempat parkir besar itu sangat sulit. Artinya, pengemudi harus berkeliraran di dalam area parkir untuk mencari tempat parkir kosong. Selain itu, diperlukan tenaga ekstra untuk mencari tempat parkir yang kosong, Sistem parkir konvensional saat ini menitik beratkan pengemudi untuk mencari lahan parkir sendiri yang dapat menambah pembuangan emisi gas dan waktu, Oleh karena itu dibutuhkan sebuah sistem untuk mencari ruang kosong yang terdapat pada tempat parkir. Pada penelitian ini penulis menggunakan metode *Convolutional Neural Networks* (CNN), dan untuk dataset yang digunakan berupa citra dari dataset publik PKLot. dalam mengenali ketersediaan lahan parkir yang kosong penulis menggunakan arsitektur AlexNet yang memiliki parameter lebih sedikit dari arsitektur lain. Hasil terbaik pada penelitian ini menunjukkan rata rata akurasi 99% dengan data uji.

Kata kunci: *Convolutional Neural Networks* (CNN), tempat parkir, PKLot.

Abstract

Parking in a large parking lot is very difficult. That is, the driver must roam the parking area to find an empty parking space. In addition, extra energy is needed to find an empty parking space, the conventional parking system currently focuses on the driver to find his own parking space that can increase the disposal of gas emissions and time, therefore we need a system to search for empty spaces contained in the parking lot In this study, the author uses the Convolutional Neural Networks (CNN) method, and for the dataset used in the form of images from the public PKLot dataset, in recognizing the availability of vacant parking lots, the author uses the AlexNet architecture which has fewer parameters than other architectures. The best results of this study showed an average accuracy of 99% with test data.

Keywords: *Convolutional Neural Networks* (CNN), parking lot, PKLot.

1. Pendahuluan

Latar Belakang

Semenjak tahun 2015 hingga sekarang meningkatnya pertambahan volume kendaraan berdampak pada kebutuhan akan sarana parkir. menurut Badan Pusat Statistik Indonesia terhitung terdapat 3.5 Juta mobil penumpang yang terdapat di Jakarta pada tahun 2018 dan angka ini terus bertambah hingga sekarang. Banyak pengemudi menghadapi kesulitan dalam menemukan tempat parkir yang tersedia pada saat berkunjung ke suatu tempat. Bahkan ketika hampir semua pusat perbelanjaan besar dan tempat-tempat umum lainnya memiliki gedung parkir bertingkat tersendiri.

Lahan parkir merupakan salah satu sarana penting yang harus dimiliki oleh sebuah tempat umum seperti perkantoran, kampus, pusat perbelanjaan, dan lain-lain. Kebanyakan lahan parkir yang ada pada tempat umum tersebut memiliki lahan yang luas, sehingga para pengendara kendaraan bermotor sulit untuk menjangkau lahan parkir tersebut satu per satu. Pencarian tempat parkir satu per satu akan membuat antrian kendaraan yang akan parkir menjadi padat dan menyebabkan kemacetan di lahan parkir. Sampai saat ini pengelola lahan parkir masih menggunakan data yang menginformasikan berapa jumlah slot parkir kosong tanpa memberikan informasi lokasi slot parkir yang kosong tersebut. Di beberapa tempat parkir sudah diterapkan dengan menggunakan alat berupa sensor gerak yang ditaruh di atas tempat parkir [1], namun terdapat kekurangan dari digunakan sensor gerak dalam mengetahui tempat parkir tersebut tersedia atau tidak. Pada sensor gerak harus dipasang disetiap lahan parkir yang ada. Dengan begitu akan membutuhkan biaya yang cukup besar jika lahan parkir sangat banyak. Pada penelitian ini penulis menggunakan alat berupa kamera sebagai pengganti sensor gerak. Dengan menggunakan kamera yang dipasangkan disalah satu sudut lahan parkir, kamera dapat mengambil citra lahan parkir dengan luas atau banyak.

Riset-riset serupa telah dilakukan dalam rangka memecahkan permasalahan tersebut, di antaranya membuat sebuah sistem yang akan mencari ruang kosong pada lahan parkir dengan menggunakan kamera sebagai alat utamanya. Pada suatu penelitian sebagian besar dilakukan dengan mengekstraksi fitur visual seperti *Scale-Invariant Feature Transform* (SIFT), *Speeded Up Robust Features* (SURF) dan *Histogram of Oriented Gradients* (HOG) [2]. Kelemahan menggunakan ekstraksi fitur adalah kemampuan terbatas fitur tersebut untuk beradaptasi

dengan variasi penampilan objek yang sangat non-linear, bervariasi waktu dan kompleks [3].

Beberapa sistem yang telah dibuat di antaranya menggunakan salah satu metode *Deep Learning* yang sangat populer dalam enam tahun terakhir yaitu CNN[4]. Pada tahun 2017 dilakukan penelitian dengan CNN sebagai metodenya dan dilatih oleh dataset citra berukuran besar. Pada penelitian tersebut menghasilkan kinerja yang luar biasa dalam kasus pencarian lahan parkir dari citra yang digunakan [5]. CNN juga dapat melihat fitur dari sebuah citra. Semakin dalam *layer* maka semakin kompleks pula fitur yang akan dipelajari [6]. Dengan melihat fitur-fitur tersebut, CNN dapat melakukan klasifikasi terhadap sebuah citra.

Pada penelitian sebelumnya AlexNet dapat mengenali 1000 kelas yang ada [6], namun pada kasus lahan parkir kelas yang dimiliki hanya 2. Oleh karena itu peneliti sedikit menyesuaikan arsitektur AlexNet agar dapat digunakan pada penelitian ini. Dengan begitu arsitektur yang semula dapat mengklasifikasikan 1000 kelas sekarang dapat mengklasifikasikan dua kelas dengan melatih model dari awal [7].

Oleh karena itu, penelitian ini membuat sebuah sistem yang dapat mengenali ruang kosong pada tempat parkir dengan menggunakan CNN.

Topik dan Batasannya

Berdasarkan latar belakang masalah penelitian ini, dapat disimpulkan bahwa topik permasalahan yang diselesaikan yaitu membangun sistem yang dapat mengenali ruang kosong pada lahan parkir menggunakan *Convolutional Neural Network* (CNN).

Batasan permasalahan pada penelitian ini yaitu:

1. Dataset yang digunakan adalah dataset publik PK10t
2. Dataset yang digunakan adalah citra lahan parkir mobil dengan format JPEG dengan resolusi 1280x720 piksel.
3. Studi kasus dilakukan di lahan parkir yang memiliki koordinat pada setiap tempat parkir.
4. Sistem yang dibangun tidak mengenali jenis mobil.
5. Skenario yang digunakan saat penelitian adalah sebagai berikut:
 - Melakukan pengenalan terhadap citra lahan parkir.
 - Pengaruh cuaca dalam kinerja mengenali lahan parkir kosong.

Tujuan

Tujuan dari penelitian ini adalah membangun sebuah sistem yang dapat mengenali ruang kosong pada lahan parkir mobil. Pada penelitian ini menggunakan metode *Convolutional Neural Network*. *Convolutional Neural Network* dapat mengekstraksi fitur menggunakan jaringan konvolusi dan dapat juga digunakan sebagai klasifikasi.

Organisasi Tulisan

Penelitian ini dibagi menjadi 5 bagian, Bagian pertama mengenai latar belakang mengapa penelitian dilakukan. Pada bagian kedua dijelaskan mengenai studi terkait dengan penelitian. Pada bagian ketiga dijelaskan sistem yang dibangun. Pada bagian keempat dijelaskan mengenai hasil dan evaluasi dari sistem yang telah dibangun. Pada bagian kelima, dibahas hasil penelitian yang dilakukan.

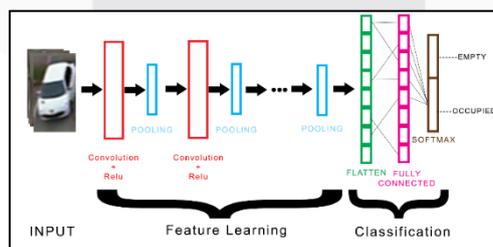
2. Studi Terkait

Convolutional Neural Network

Algoritma *Convolutional Neural Network* (CNN) adalah algoritma yang umum digunakan pada bidang *Computer Vision* untuk mengetahui berbagai macam jenis obyek pada citra. Algoritma ini mampu melakukan segmentasi untuk mengetahui dimana obyek tersebut berada[8]. Banyak sekali penelitian yang telah menunjukkan hasil yang memuaskan dengan menggunakan algoritma ini.

Sebuah CNN biasanya memiliki *input layer*, *output layer* dan *hidden layers*. Pada CNN *hidden layers* terdiri dari *convolutional layers*, *pooling layers*, *fully connected layers* (FC layer), dan *normalization layers*[8].

Input untuk algoritma ini adalah data citra atau sebuah array dua dimensi dengan tiga channel warna. Sedangkan output berupa hasil prediksi dari data yang digunakan sebagai input. Contoh kasus adalah sebuah citra kucing akan digunakan sebagai input untuk memprediksi citra tersebut masuk ke dalam kelas kucing atau bukan, pada penelitian sebelumnya[5][7] menggunakan CNN dengan arsitektur AlexNet.



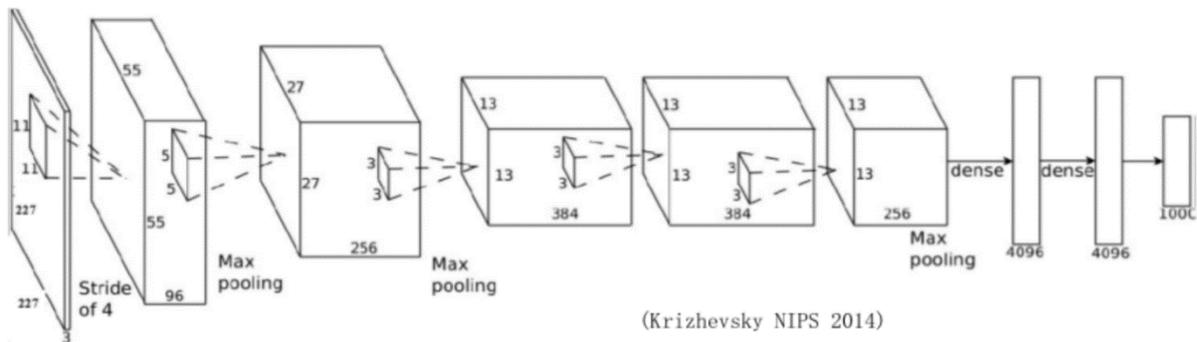
Gambar 1. Contoh arsitektur CNN

Konvolusi pada dasarnya berupa hasil matematika antara filter dengan citra. Hasil matematika ini nantinya akan digunakan untuk layer selanjutnya. Agar proses konvolusi dapat dilakukan pada sebuah citra diperlukan *stride*, kegunaan *stride* membuat proses konvolusi dapat bergeser hingga nilai akhir dari sebuah citra. Setelah itu akan menghasilkan *Feature Map* yang akan digunakan untuk *layer* selanjutnya.

AlexNet

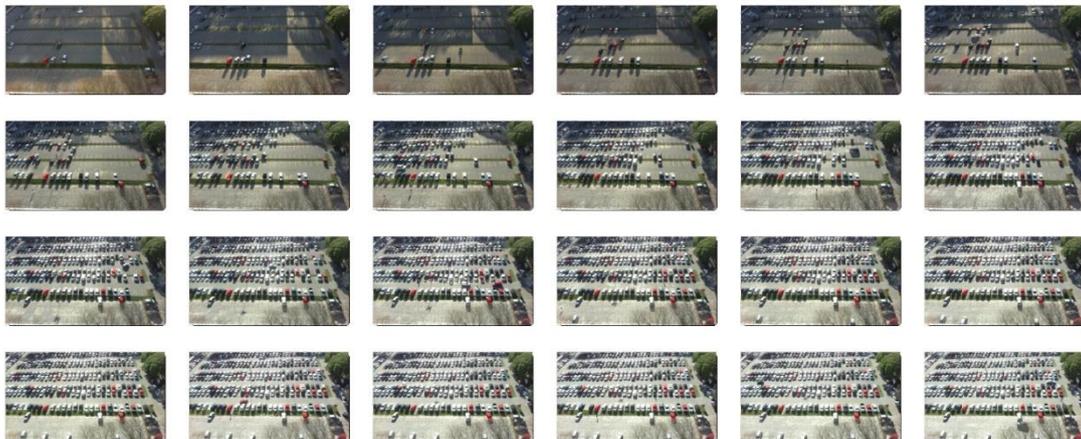
AlexNet merupakan salah satu arsitektur CNN yang banyak digunakan dalam beberapa penelitian. AlexNet merupakan arsitektur pertama yang menerapkan Relu sebagai fungsi aktivasi di setiap layer yang ada[6]. dengan memiliki parameter sebanyak 61 juta, arsitektur ini dapat mengklasifikasi sebanyak 1000 kelas. arsitektur ini dibuat oleh Alex Krizhevsky, Ilya Sutskever dan Geoffrey E. Hinton [6].

AlexNet dibuat dengan tujuan untuk mengklasifikasikan 1,2 juta citra resolusi tinggi dalam kontes ImageNet LSVRC-2010 ke dalam 1000 kelas yang berbeda [6]. Pada data uji mendapatkan kesalahan dari 37.5% dan 17.0% yang jauh lebih baik daripada keadaan sebelumnya. Arsitektur ini memiliki 61 juta parameter dan 650.000 neuron, terdiri dari lima lapisan konvolusional, beberapa di antaranya diikuti oleh lapisan *Max Pooling*, dan tiga lapisan yang terhubung sepenuhnya dengan *softmax*.



Gambar 2. Ilustrasi Arsitektur AlexNet[8]

PKLot



Gambar 3. Contoh beberapa citra pada PKLot

PKLot merupakan dataset lahan parkir yang tersedia pada tahun 2015 dan masih tersedia dengan tujuan penelitian, dataset ini diperoleh ditempat tempat parkir Universitas Federal Parana (UFPR) dan Universitas Katolik Kepasuan Parana (PUCPR). Keduanya berlokasi di Curitiba, Brasil. Dengan total 12.417 citra tempat parkir dan 695.899 citra ruang parkir yang disegmentasi darinya, yang diperiksa dan diberi label secara manual [9].

Proses ini didefinisikan untuk dieksekusi dengan interval selang waktu 5 menit. Dan diambil lebih dari 30 hari, dengan menggunakan kamera definisi tinggi (Microsoft LifeCam, HD-5000) yang diposisikan di bagian atas gedung. Oleh karena itu meminimalkan kemungkinan kerapatan antara kendaraan yang berdekatan. Tujuan utama adalah untuk mendapatkan citra dalam kondisi cuaca yang berbeda (mendung, cerah, dan periode hujan) dengan mengambil pada setiap 5 menit perubahan lingkungan[9].

Pada penelitian ini citra yang peneliti gunakan memiliki 100 lahan parkir dalam 1 citra, dengan deksripsi setiap cuaca pada tabel 1.

Lahan parkir	Kondisi Cuaca	Lama hari	Banyaknya gambar	Ruang Parkir		
				Terisi	Kosong	Total
PUCPR (100 Tempat Parkir)	Cerah	24	2315	95.161 (46.42%)	111.672 (53.58%)	206.833
	Mendung	11	1328	41.521 (31.90%)	90.417 (68.10%)	131.938
	Hujan	8	831	55.104 (66.35%)	27.959 (33.65%)	83.063
	Total		4474	191.786 (45.78%)	230.048 (54.22%)	421.834

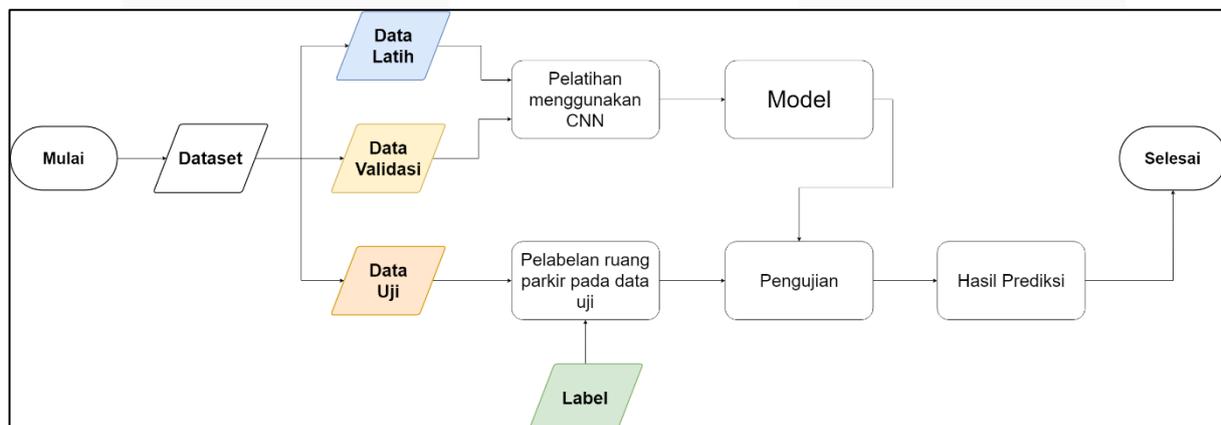
Tabel 1. Deskripsi pada PKLot

Untuk setiap citra tempat parkir dibuat file *Extensible Markup Language* (XML) yang berisi posisi dan situasi (kosong atau ditempati) dari setiap tempat parkir. Dengan menggunakan XML dapat memberi label pada citra. Label ini memungkinkan visualisasi setiap citra dan definisi batas setiap ruang parkir (diwakili oleh titik-titik poligon), serta situasi (kosong atau ditempati). *Subfolder* yang berbeda digunakan untuk mengelompokkan citra secara manual sesuai dengan kondisi cuaca yang diamati (mendung, cerah, atau hujan)[9].

Setiap ruang parkir akan diekstrasi dari setiap citra menggunakan informasi label dalam file XML. Dan dengan hasil dari segmentasi yang didapat dihasilkan rincian dari setiap dataset PKLot yang didapat.

3. Sistem yang Dibangun

Pada penelitian ini, sistem dibagi menjadi tiga tahap, yaitu tahap praproses, *training* dan *testing*, dan pengukuran performa. Pada tahap praproses dilakukan *resize* pada citra dari label yang telah tersedia. Lalu pada tahap *training* dan *testing* menggunakan CNN. Terakhir pada tahap pengukuran performa akan diukur dengan beberapa formula. Gambar 4 merupakan rancangan sistem secara keseluruhan yang dibangun.



Gambar 4. Blok proses sistem

Diagram alir dari sistem yang dibangun dapat dilihat pada gambar 4. Proses pertama yang akan dilakukan terlebih dahulu adalah membagi dataset PKlot menjadi 3 bagian, yaitu menjadi data latih data validasi dan data uji. Pada penelitian ini pembagian data dilakukan secara *manual* atau dengan sendiri. Pada penelitian ini untuk data latih sebesar 60%, dan data validasi sebesar 10%, sedangkan untuk data uji sebesar 30%. Untuk data latih dan data validasi akan digunakan pada proses pembuatan model CNN, sedangkan data uji akan digunakan untuk tahap uji atau pengenalan lahan parkir.

Pada penelitian ini dibangun sebuah model CNN dengan arsitektur AlexNet, ukuran untuk *input layer* pada penelitian ini sebesar 71x71. ukuran tersebut merupakan ukuran dataset pklot yang sudah terpotong atau *crop*. Pada penelitian ini jumlah *layer* dan fungsi aktivasi mengikuti arsitektur AlexNet [6]. Pada AlexNet terdapat 5 *layer* konvolusi dan fungsi aktivasi menggunakan *ReLU* [10], namun pada *output layer/classifier* diganti dengan beberapa *layer* untuk menyesuaikan kasus pada penelitian ini.

Setelah jaringan fitur ekstraksi pada AlexNet selesai, kemudian disambungkan dengan satu *layer Flatten* yang bertujuan untuk meringkas *feature map*, kemudian pada tahap terakhir dalam pembangunan arsitektur jaringan disambungkan dengan output *layer* berisi 2 *neuron* sebagai *layer* klasifikasi. *Layer* terakhir pada penelitian ini menggunakan fungsi aktivasi *softmax* [11], Arsitektur jaringan pada penelitian ini dapat dilihat pada tabel 2.

Nama	Input size	Output size
Layer Input	71 x 71 x 3	71 x 71 x 3
Layer Konvolusi 1	71 x 71 x 3	71 x 71 x 16
Layer Max Pooling 1	71 x 71 x 16	35 x 35 x 16
Layer Konvolusi 2	35 x 35 x 16	35 x 35 x 32
Layer Max Pooling 2	35 x 35 x 32	17 x 17 x 32
Layer Konvolusi 3	17 x 17 x 32	17 x 17 x 64
Layer Konvolusi 4	17 x 17 x 64	17 x 17 x 64
Layer Konvolusi 5	17 x 17 x 64	17 x 17 x 64
Layer Max Pooling 3	17 x 17 x 64	8 x 8 x 64
Layer Flatten	8 x 8 x 64	4096
Layer Fully Connected	4096	32
Layer Fully Connected	32	16
Layer Output	16	2

Tabel 2. Arsitektur Jaringan CNN

Pada *layer* konvolusi dan *layer max pooling* menggunakan fungsi aktivasi ReLu (*Rectified Linear Units*) [10], dengan formula seperti berikut.

$$F(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (1)$$

Dari formula 1 dapat dilihat bahwa x merupakan hasil konvolusi nilai citra dengan nilai kernel. Lalu fungsi ReLu akan membuat nilai negatif menjadi 0. Dengan menggunakan ReLu membuat operasi konvolusi selanjutnya tidak gagal. Fungsi aktivasi ReLu sudah biasa digunakan pada arsitektur CNN lainnya.

Untuk *Layer output* menggunakan fungsi *softmax*[11]. Fungsi *softmax* menghasilkan nilai kemungkinan untuk kedua label "Empty" atau label "Occupied".

Praproses

Penggunaan ukuran input *layer* pada Alexnet membuat dilakukan praproses terlebih dahulu. Praproses yang dilakukan yaitu *re-size* piksel dataset. Ukuran citra pada dataset PKLot *re-size* keukuran 71x71 piksel.

Pelatihan

Pada proses pelatihan menggunakan beberapa parameter, yaitu *optimizer* menggunakan SGD (*Stochastic Gradient Descent*) [12] dikarenakan pada penelitian serupa, didapatkan *optimizer* SGD mendapatkan hasil yang paling baik[5]. Lalu untuk fungsi *loss* menggunakan *binary crossentropy*[13], dikarenakan label yang dimiliki pada penelitian ini hanya 2 atau *binary*.

Pada proses pelatihan dilakukan validasi akurasi pada setiap *epoch* menggunakan data validasi. *Monitoring* dilakukan pada nilai *validation loss* untuk mengetahui kualitas pelatihan. Jika terjadi penurunan nilai *loss validation* maka bobot secara otomatis disimpan, namun jika terjadi peningkatan nilai *loss validation* atau tidak ada perubahan nilai *loss validation*. Dalam beberapa *epoch* berturut-turut maka *learning rate* akan otomatis diupdate dari *learning rate* sebelumnya. Pelatihan akan dihentikan ketika terjadi dalam 10 kali perulangan berturut berturut nilai akurasi validasi tidak berubah.

Spesifikasi Sistem

Dalam melakukan penelitian diperlukan juga sebuah perangkat yang dapat menunjang penelitian ini, berikut perangkat yang digunakan dalam penelitian ini:

- Sistem Operasi : Windows 10
- RAM: 16 GB
- Processor : Intel Core i7-8750H
- Bahasa Pemrograman : Python

Pengujian

Dalam penelitian ini dilakukan 2 (dua) skenario pengujian pada sistem, yaitu:

1. Skenario-1

Pada Skenario-1 melakukan pengenalan citra lahan parkir. Tujuan dari skenario ini adalah untuk mengetahui dari lahan parkir yang ada, dimanakah terdapat lahan parkir yang tersedia, lalu menghitung akurasi kinerja model CNN yang telah dibangun.

2. Skenario-2

Pada Skenario-2 melakukan pengenalan citra lahan parkir saat cuaca cerah, hujan dan berawan. Tujuan dari skenario ini adalah dapat mengetahui performansi sistem pada setiap cuaca yang ada.

Pengukuran Kinerja

		Aktual	
		Positif	Negatif
Prediksi	Positif	<i>True Positive</i>	<i>False Positive</i>
	Negatif	<i>False Negative</i>	<i>True Negative</i>

Tabel 3. Confusion Matrix

Pengukuran kinerja menggunakan *confusion matrix* [14]. *Confusion matrix* merupakan metode yang sering digunakan untuk mengukur akurasi, *precision*, *recall*, *Specificity* dari suatu sistem klasifikasi.

Terdapat 4 istilah untuk merepresentasikan hasil klasifikasi yaitu *True Positive* (TP), True Negatif (TN), False Positive (FP), dan False Negative (FN) seperti pada Tabel 3, pada *True Positive* (TP) merupakan jumlah tempat parkir kosong yang berhasil dikenali. *True Negative* (TN) merupakan jumlah tempat parkir terisi yang berhasil dikenali. *False Positive* (FP) merupakan jumlah tempat parkir kosong yang tidak berhasil dikenali. *False Negative* (FN) merupakan jumlah tempat parkir terisi yang tidak berhasil dikenali.

Selanjutnya formulasi untuk menghitung akurasi, *precision*, *recall*, *specificity* terdapat pada persamaan dibawah.

$$akurasi = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (2)$$

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

$$Specificity = \frac{TN}{TN+FP} \quad (5)$$

Pada penelitian ini, pengukuran kinerja dilakukan untuk mengevaluasi sistem yang telah dibangun. Kinerja sistem dihitung berdasarkan TP, FP, TN, FN dari setiap citra uji.

Formula yang digunakan pada penelitian ini bertujuan sebagai berikut, untuk nilai akurasi agar dapat mengetahui berapa persen lahan parkir yang benar diprediksi kosong dan tidak kosong dari keseluruhan lahan parkir pada citra. lalu nilai *precision* dapat mengetahui berapa persen lahan parkir yang benar kosong dari keseluruhan lahan parkir yang diprediksi kosong. Lalu nilai *Recall* untuk mengetahui persentase lahan parkir yang diprediksi kosong dari keseluruhan lahan parkir yang sebenarnya kosong. yang terakhir nilai *specificity* agar mengetahui persentase lahan parkir yang benar diprediksi terisi dari keseluruhan lahan parkir yang sebenarnya terisi.

4. Evaluasi

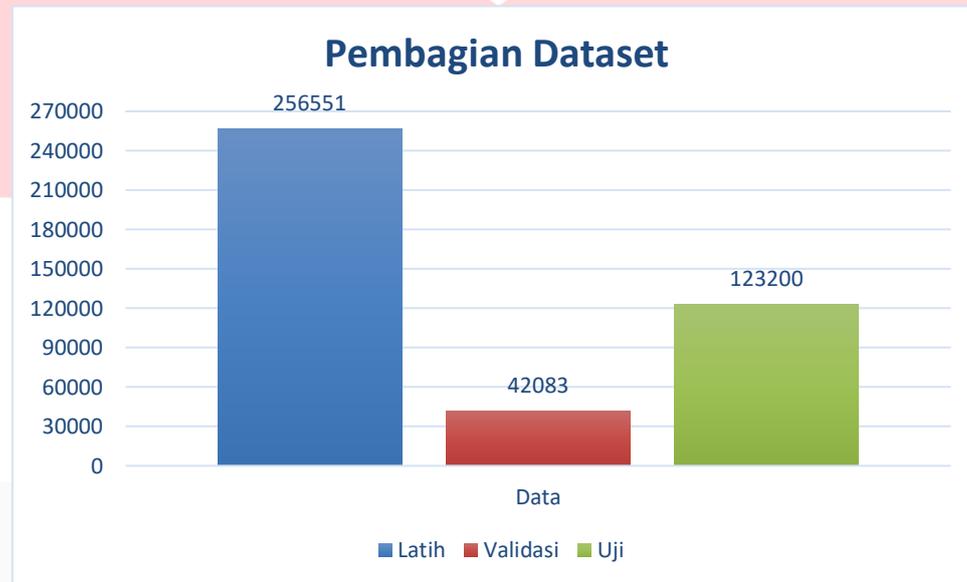
Pada tugas akhir ini, untuk menilai seberapa baik model CNN yang dipakai, implementasi model dilakukan dengan beberapa skenario uji, pada tugas akhir ini penulis merumuskan terdapat 2 skenario uji yaitu:

- Melakukan pengenalan citra lahan parkir kosong.
- Pengaruh cuaca dalam kinerja pengenalan lahan parkir kosong.

Setiap model dilatih menggunakan *optimizer SGD (Stochastic Gradient Descent)* dikarenakan pada penelitian J. Nyambal dan R. Klein [5] didapatkan SGD sebagai hasil paling tinggi. Lalu untuk nilai momentum sebesar 0.9. *Learning rate* sebesar 0.0001. Dilatih sebanyak 15 *epoch* dan setiap *epoch* nya dilatih dengan *batch* sebesar 64. Setiap model CNN dilatih menggunakan *Processor Intel Core i7-8750H*.

4.1 Hasil Pengujian

Pada tahap ini pembagian data latih dan data uji juga menjadi pertimbangan penulis, oleh karena itu untuk pembagian yang digunakan penulis pada penelitian ini dapat dilihat sebagai berikut.



Gambar 5. Pembagian Dataset Skenario 1

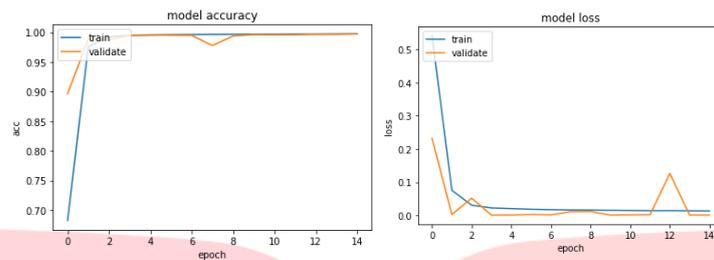
Dapat dilihat pada Gambar 5, penulis membagi setiap data dengan presentase data latih sebesar 60% , data validasi sebesar 10% dan data uji sebesar 30%. Untuk label pada penelitian ini terdapat 2 label dengan rincian sebagai berikut.

Tipe Data	Label		Total
	Kosong (empty)	Terisi (occupied)	
Data Latih	140.692	115.859	256.551
Data Validasi	15.478	26.605	42.083
Data Uji	73.838	49.362	123.200
	Total		421.834

Tabel 4. Rincian label yang digunakan Skenario 1

Pengenalan citra lahan parkir

Dalam skenario uji ini, bertujuan untuk melakukan pengenalan pada lahan parkir yang tersedia dalam citra masukan. Untuk arsitektur CNN yang digunakan adalah arsitektur AlexNet, dengan rincian terdapat pada tabel 2. untuk pembagian data latih dan uji terdapat pada Gambar 5 .



Gambar 6. Grafik Pelatihan Skenario Pertama

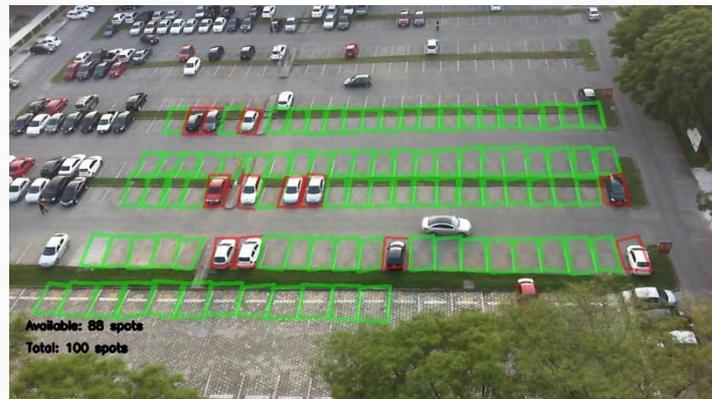
Pada Gambar 6 adalah grafik dari hasil pelatihan yang dilakukan pada penelitian ini. Pelatihan dilakukan sebanyak 15 *epoch*. Lalu didapat akurasi *train validation* sebesar 99,67%, namun jika melihat kembali pada Gambar 6 terdapat tanda-tanda *overfitting* pada *epoch* ke 2. Ditandai dengan grafik garis akurasi dan *loss* dari data validasi meningkat daripada data latih. Sehingga proses pelatihan hanya menyimpan model terbaik sebelum *overfitting*. Dalam proses latih ini memerlukan 1200 detik untuk setiap *epoch* nya, jika ditotalkan dengan 15 *epoch* memerlukan kurang lebih waktu 5 jam.

Setelah tahap latih selesai dilakukan hasil akhir berupa model yang sudah dilatih, lalu masuk ke tahap uji dengan menggunakan *Confusion Matrix* sebagai metode ukur perfomansi sistem.

Accuracy	Precision	Recall	Specificity
99,05%	96,94%	96,98%	82,76%

Tabel 5. Hasil Uji Skenario Pertama

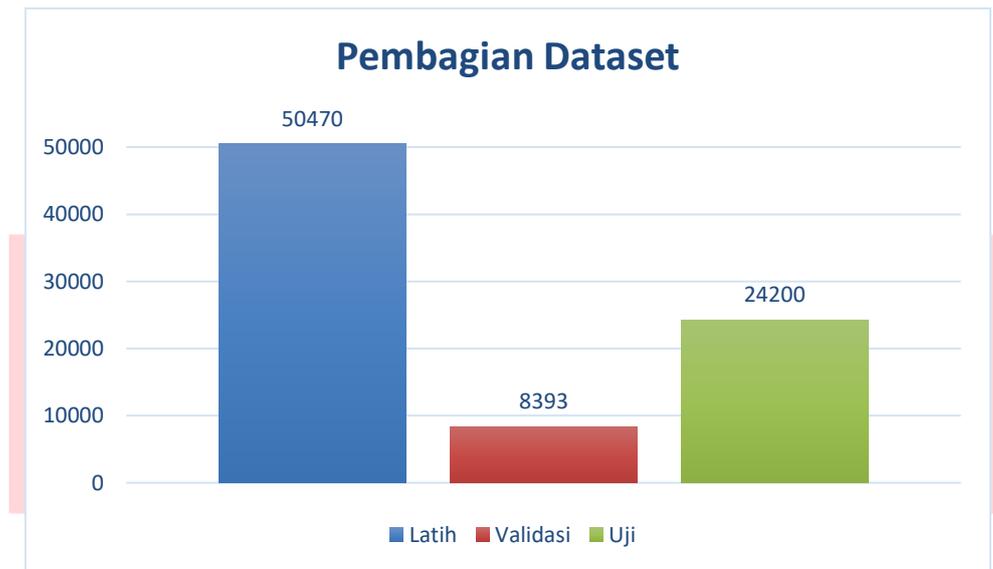
Pada Tabel 5 ditampilkan hasil dari observasi yang telah dilakukan dengan menggunakan arsitektur CNN Tabel 2. Pada tabel 5 dapat dilihat mendapatkan peningkatan dari penelitian[5] sebelumnya yang menggunakan CNN dengan arsitektur AlexNet.



Gambar 7. Contoh hasil uji

Pengaruh cuaca dalam kinerja mengenali lahan parkir kosong

Dalam skenario uji ini bertujuan untuk membandingkan kinerja sistem pada cuaca cerah, berawan dan hujan. Untuk arsitektur CNN yang digunakan merupakan seperti pada Tabel 2, dalam pembagian data uji, data validasi dan data latih dapat dilihat pada gambar 8.

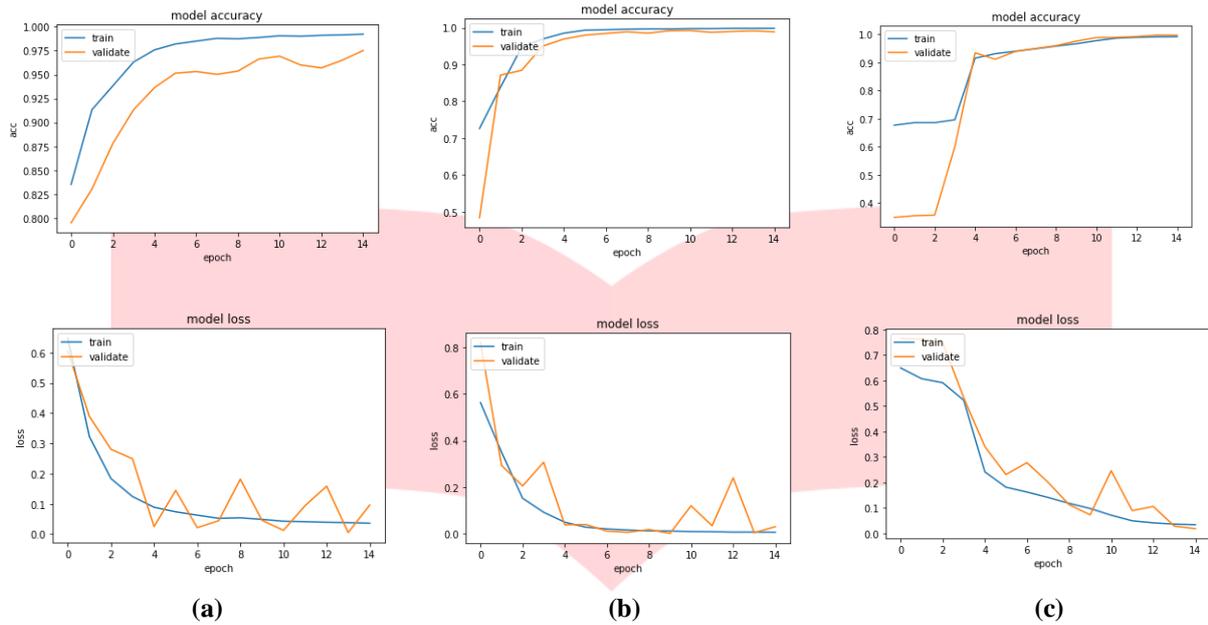


Gambar 8. Pembagian Dataset Skenario 2

Dapat dilihat pada Gambar 8 pada penelitian ini penulis membagi setiap data dengan presentase data latih sebesar 60% , data validasi sebesar 10% dan data uji sebesar 30%. Untuk label pada skenario uji ini terdapat 2 label dengan rincian sebagai berikut.

Cuaca	Tipe Data	Label		Total
		Kosong (empty)	Terisi (occupied)	
Cerah	Data Latih	25.235	25.235	50.470
	Data Validasi	4.203	4.190	8.393
	Data Uji	18.786	5.414	24.200
	Total			83.063
Berawan	Data Latih	36.649	13.821	50.470
	Data Validasi	4.039	4.354	8.393
	Data Uji	12.703	11.497	24.200
	Total			83.063
Hujan	Data Latih	15.856	34.614	50.470
	Data Validasi	5.441	2.952	8.383
	Data Uji	6.661	17.539	24.200
	Total			83.063

Tabel 6. Rincian label yang digunakan Skenario 2



Gambar 9. Grafik Pelatihan Skenario Kedua (a) Cerah (b) Berawan (c) Hujan

Pada Gambar 9 adalah grafik yang didapatkan dari proses pelatihan. Pelatihan dilakukan dalam 15 *epoch*, dan didapatkan *train validation* sebesar 99%. Namun jika dilihat kembali pada Gambar 9 terdapat tanda-tanda *overfitting* pada setiap cuaca. Yang ditandai dengan grafik garis akurasi dan *loss* dari data validasi meningkat daripada data latih. Sehingga proses pelatihan hanya menyimpan model terbaik sebelum *overfitting*, untuk waktu latihnya dapat dilihat di tabel berikut.

Cuaca	Waktu (s)	Ubah ke Menit (m)	Ubah Ke Jam(h)
Cerah	4020	67	1.116
Berawan	4124	68	1.14
Hujan	4329	72.15	1.2025

Tabel 7. Waktu Komputasi untuk Proses Latih Skenario Kedua

Pada Tabel 7 terdapat perbedaan waktu yang tidak terlalu berbeda pada setiap cuaca yang ada, itu disebabkan karena jumlah data latih yang ada pada setiap cuaca sama.

Setelah proses latih selesai dilanjutkan ke proses uji dengan rincian data seperti Gambar 8, untuk pengukuran peromansi uji menggunakan *Confusion Matrix*, dengan waktu komputasi yang didapatkan sebagai berikut.

Cuaca	Waktu (s)	Ubah ke Menit (m)
Cerah	107	1.78
Berawan	108	1.8
Hujan	109	1.81

Tabel 8. Waktu Komputasi untuk Proses Uji Skenario Kedua

Pada Tabel 8 setiap cuaca memiliki perbedaan waktu komputasi yang tidak terlalu berbeda, namun untuk waktu rata rata setiap citra ujinya adalah 1 detik.

Perfomansi	Cuaca		
	Cerah	Berawan	Hujan
Accuracy	94,92%	98,50%	97,80%
Precision	93,55%	95,73%	92,80%
Recall	99,67%	88,61%	91,22%
Specifity	69,87%	71,08%	96,90%

Tabel 9. Hasil Uji Skenario Kedua

Pada Tabel 9 didapati cuaca berawan merupakan kondisi yang sangat prima untuk sistem yang telah dibangun. Pada cuaca hujan intensitas cahaya yang diambil sangat gelap dan beberapa citra terdapat *Noise*. *Noise* tersebut yang dapat menjadi penyebab penurunan perfomansi. Pada cuaca cerah bayangan yang terdapat pada citra juga dapat mempengaruhi hasil perfomansi sistem.



Gambar 10. Contoh Data Uji pada Cuaca Cerah dan Hujan

5. Kesimpulan

Pada penelitian ini, telah dibangun sistem pengenalan lahan parkir kosong untuk lahan parkir mobil dengan menggunakan *Convolutional Neural Network* (CNN). Berdasarkan penelitian yang telah dilakukan dapat disimpulkan bahwa CNN dapat melakukan pengenalan pada lahan parkir mobil. Dengan menggunakan arsitektur AlexNet akurasi didapat sebesar 99,05%. Lalu terdapat perbedaan hasil perfomansi yang dialami jika berbeda cuaca. dalam cuaca berawan mampu menghasilkan nilai akurasi, *precision*, *recall* dan *specificity* yang cukup baik. Dengan menggunakan data yang sama persis dengan penelitian yang sudah ada, hasil dari akurasi pada penelitian ini yaitu 99,05%. Hal ini mengalami peningkatan daripada penelitan sebelumnya.

Pada penelitian ini, observasi pada dataset dan *hyperparameter* sangat sedikit dilakukan, meskipun demikian mampu mencapai peningkatan akurasi rata-rata yang signifikan dari penelitian sebelumnya. Diharapkan pada penelitian selanjutnya dapat dilakukan observasi pada dataset dan *hyperparameter* agar didapatkan parameter yang lebih optimal. Sehingga mampu meningkatkan akurasi saat terdapat kondisi hujan deras atau saat kondisi kurang pencahayaan.

Daftar Pustaka

- [1] S. Lee, D. Yoon, and A. Ghosh, "Intelligent parking lot application using wireless sensor networks," *2008 Int. Symp. Collab. Technol. Syst. CTS'08*, pp. 48–57, 2008.
- [2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 580–587, 2014.
- [3] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, 2006.
- [4] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.
- [5] J. Nyambal and R. Klein, "Automated parking space detection using convolutional neural networks," *2017 Pattern Recognit. Assoc. South Africa Robot. Mechatronics Int. Conf. PRASA-RobMech 2017*, vol. 2018-Janua, pp. 1–6, 2017.
- [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [7] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, "Deep learning for decentralized parking lot occupancy detection," *Expert Syst. Appl.*, vol. 72, pp. 327–334, 2017.
- [8] S. Albawi, T. A. M. Mohammed, and S. Alzawi, "Understanding of a Convolutional Neural Network," *Ieee*, 2017.
- [9] P. R. L. De Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich, "PKLot-A robust dataset for parking lot classification," *Expert Syst. Appl.*, vol. 42, no. 11, pp. 4937–4949, 2015.
- [10] V. Nair and G. E. Hinton, "Rectified linear units improve Restricted Boltzmann machines," in *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 2010, pp. 807–814.
- [11] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-Margin Softmax Loss for Convolutional Neural Networks," 2016.
- [12] S. Ruder, "An overview of gradient descent optimization algorithms," pp. 1–14, 2016.
- [13] Y. Ho and S. Wookey, "The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020.
- [14] D. Visa Sofia, "Confusion Matrix-based Feature Selection Sofia Visa," *ConfusionMatrix-based Featur. Sel. Sofia*, vol. 710, no. January, p. 8, 2011.