

BAB II

LANDASAN TEORI

2.1 Verifikasi Sistem Parkir

Verifikasi sistem parkir merupakan hal yang sangat penting karena keamanan yang sangat diperlukan saat berada di lingkungan area parkir. Kemananan di lingkungan area parkir menjadi sebuah perhatian khusus untuk dilakukan penelitian ini. Verifikasi dilakukan untuk mendapatkan suatu kebenaran atau kepastian setelah dilakukannya pemeriksaan, maka dari itu verifikasi pada sistem parkir merupakan permasalahan yang penting untuk diteliti. Objek dalam penelitian ini adalah merujuk pada sistem verifikasi parkir pada kendaraan roda dua. Seperti halnya verifikasi parkir yang terjadi di lingkungan kampus IIB Darmajaya Lampung, dimana verifikasi sistem parkir dilakukan dengan menunjukkan STNK kepada petugas keamanan kampus, saat kendaraan roda dua keluar dari area parkir kampus.

Verifikasi sistem parkir sudah dilakukan sejak lama di lingkungan kampus dan tidak selalu dilakukan secara berkala. Verifikasi sistem parkir yang berlaku sekarang ini juga terkadang menimbulkan masalah antrian yang panjang saat jumlah kendaraan banyak yang keluar dari area parkir sehingga menimbulkan kepadatan di area keluar masuknya kendaraan. Verifikasi pada sistem parkir juga dilakukan untuk mengurangi tingkat kejahatan pada kendaraan roda dua yang sering terjadi di lingkungan area kampus.

2.2 *Computer Vision*

Computer vision merupakan salah satu subdisiplin dari ilmu kecerdasan buatan yang akan membuat komputer mampu melakukan sebuah verifikasi pada pengendara dan kendaraannya saat hendak masuk dan keluar dengan cepat. Menurut Jurnal Pseudocode oleh (Prabowo, Abdullah and Manik, 2018)

Computer Vision adalah untuk menduplikasi kemampuan penglihatan manusia ke dalam benda elektronik sehingga benda elektronik dapat memahami dan mengerti arti dari gambar yang dimasukkan. *Computer vision* berusaha menerjemahkan citra menjadi deskripsi atau suatu informasi yang mempresentasikan citra tersebut, sehingga proses *computer vision* menggunakan citra sebagai input sehingga menghasilkan output berupa informasi.

Menurut Jurnal Teknik Informatika oleh (Wibowo, 2016) Tujuan utama dari *Computer Vision* adalah agar komputer atau mesin dapat meniru kemampuan perseptual mata manusia dan otak, atau bahkan dapat mengunggulinya untuk tujuan tertentu. Kemampuan *computer vision* yang dapat mengenali serta mempresentasikan citra atau objek menjadi sumber informasi inilah yang menjadi sebuah sub disiplin ilmu yang sangat sesuai digunakan dalam penelitian ini yang dibutuhkan sebuah kemampuan ilmu kecerdasan buatan yang mampu mengenali objek sehingga dapat menverifikasinya.

2.3 Image Preprocessing

Tahapan awal proses penelitian ini setelah pengumpulan data adalah *image preprocessing*. *Preprocessing* merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada gambar input untuk proses selanjutnya. Menurut Jurnal *Journal of Computer Science and Information Systems*, oleh (Rizarta and Avianto, 2019) *Preprocessing* adalah proses awal dilakukannya perbaikan citra sebelum dilakukan proses lebih lanjut terhadap citra masukan. *Preprocessing* bertujuan supaya meningkatkan kemungkinan keberhasilan pada tahap pengolahan lebih lanjut pada sebuah citra. *Preprocessing* merupakan suatu proses untuk menghilangkan bagian-bagian yang tidak diperlukan pada gambar input untuk proses selanjutnya. *Preprocessing* dilakukan agar tidak mengganggu proses pengolahan citra terutama pada citra *template* penelitian. *Preprocessing* juga akan membantu memperjelas bagian yang menjadi target dalam objek penelitian. *Preprocessing* bertujuan untuk memperbaiki kualitas citra sebelum dilakukannya segmentasi citra.

2.3.1 Cropping

Menurut Jurnal Teknik Informatika dan Sistem Informasi oleh (Saifullah, Yudhana and Sunardi, 2016) *Cropping* merupakan suatu proses untuk memperkecil ukuran citra dengan memotong citra pada koordinat tertentu pada area citra. Sehingga proses *cropping* akan memotong sebagian objek pada bidang gambar untuk mendapatkan bagian dari gambar tertentu dengan ukuran tertentu. *Preprocessing* merupakan tahapan yang bertujuan untuk mendapatkan *Region Of Interest* (ROI) dengan cara memotong citra (*cropping*).

2.4 Segmentasi Citra

Segmentasi merupakan proses setelah dilakukannya *preprocessing*. Tujuan segmentasi adalah mendapatkan representasi sederhana dari suatu citra sehingga lebih mudah dalam pengolahannya. Segmentasi citra dilakukan untuk mendapatkan hasil citra yang diinginkan dalam melakukan proses pengolahan citra selanjutnya. Menurut Jurnal Information System Development oleh (Pangaribuan, 2019) Segmentasi Citra dalam visi komputer, adalah proses mempartisi citra digital menjadi beberapa segmen. Tujuan dari segmentasi adalah untuk menyederhanakan dan atau mengubah penyajian gambar ke sesuatu yang lebih bermakna. Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek-objek yang terkandung di dalam citra atau membagi citra ke dalam beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut.

2.4.1 Region of Interest (ROI)

Hasil yang didapat dari *preprocessing* adalah *cropping* sehingga mendapatkan daerah yang menjadi target atau area yang dibutuhkan dalam proses pengolahan segmentasi yaitu *Region of Interest* (ROI). Menurut Jurnal Teknologi Informasi dan Ilmu Komputer oleh (Pratomo, Kaswidjanti dan Mu'arifah, 2020) *Region of Interest* (ROI) merupakan salah satu teknik segmentasi sebagai proses pengolahan citra dimana pengguna mampu

mengolah citra yang mengandung informasi data citra yang dikehendaki. ROI bekerja dalam pengkodean secara berbeda pada area tertentu dari citra digital sehingga area citra yang lebih penting akan memiliki kualitas yang lebih baik dari area disekitarnya.

2.4.2 Titik kunci (*Keypoint*)

Segmentasi yang dilakukan untuk memproses citra dalam penelitian ini yaitu menggunakan titik kunci (*keypoint*). Dalam Jurnal Komputasi oleh (Harahap, 2016) menyatakan *keypoint* adalah titik-titik dari sebuah gambar yang nilainya tetap ketika mengalami perubahan skala, rotasi, *blurring*, pencahayaan, dan juga perubahan bentuk. Perubahan bentuk ini bisa terjadi karena bentuk gambar *query* yang tidak utuh atau tidak sesempurna gambar yang ada didalam *database* gambar tersebut.

2.5 *Template Matching*

Dalam penelitian ini dilakukan pendekatan dengan menggunakan metode *template matching* yang proses dimana citra dalam penelitian ini akan dicocokkan dengan berkas citra yang sudah diambil dan disimpan sebagai *template*. Citra input akan dicocokkan dengan mencari kesamaan yang ada pada *cdatabase* citra *template*. Jika ditemukan kemiripan yang hampir sama dengan *template* yang sudah ada, hal tersebut menandakan adanya kecocokan dan sebaliknya jika tidak ada banyak kemiripan hal tersebut diartikan bahwa citra input tidak *match* dengan *template* yang ada, sehingga penelitian ini menggunakan *template matching* sebagai metodenya.

Menurut Jurnal Technologia oleh (Rosyadi, 2017) *Template matching* adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi *template* (acuan). Citra masukan dibandingkan dengan citra *template* yang ada didalam basis data, kemudian dicari kesamaannya dengan menggunakan suatu aturan tertentu. Pencocokan citra yang menghasilkan tingkat kemiripan / kesamaan yang tinggi menentukan suatu citra tersebut dikenali sebagai salah satu dari citra

template. Perbandingan antara *template* dengan keseluruhan objek pada citra dapat dilakukan dengan menghitung selisih jarak, seperti berikut :

$$D(m, n) = \sum_j \sum_k [f(j, k) - T(j - m, k - n)]^2$$

Dengan $f(j,k)$ menyatakan citra tempat objek yang akan dibandingkan dengan *template* $T(j,k)$ sedangkan $D(m,n) = 0$, namun kondisi ini sulit dipenuhi apalagi bila *template* merupakan suatu citra *grayscale*s. Oleh karena itu, aturan yang digunakan untuk menyatakan *template* cocok dengan objek bila :

$$D(m, n) < L_D(m, n)$$

Dengan $L_D(m, n)$ merupakan nilai *threshold*.

2.5.1 Algoritma SIFT (*Scale Invariant Feature Transform*)

Pencocokan citra yang dilakukan menggunakan algoritma SIFT. Dalam Jurnal CyberSecurity dan Forensik Digital oleh (Rosidin, Sugiantoro dan Prayudi, 2018) Algoritma SIFT adalah algoritma dalam visi komputer untuk mendeteksi dan menggambarkan fitur lokal pada image atau digunakan untuk mencocokkan gambar berdasarkan fitur *keypoint* utama (*invariant* skala dan rotasi). Algoritma SIFT adalah ekstraksi fitur yang paling banyak digunakan untuk menemukan titik-titik kunci pada image yang akan dicocokkan.

2.5.2 *K-Nearest Neighbor* (KNN)

Keypoint yang ada pada citra input dicocokkan dengan citra yang ada dalam *database* yaitu dengan mencari jumlah kesamaan titik yang paing banyak atau mendekati dengan citra yang ada, hal ini diperlukan metode KNN untuk mencari kesamaan tedekat antar citra. Menurut Jurnal Online Informatika oleh (Syahid, Jumadi dan Nursantika, 2016) KNN (*K-Nearest Neighbor*) adalah metode untuk melakukan klasifikasi terhadap objek berdasarkan data pembelajaran yang jaraknya paling dekat atau memiliki persamaan ciri paling banyak dengan objek tersebut. Jarak yang menentukan kedekatan antar citra dalam penelitian ini digunakan jarak

Euclidean. Menurut Jurnal Pengembangan IT (JPIT) oleh (Nishom, 2019) Jarak *Euclidean* merupakan merupakan salah satu metode perhitungan jarak yang digunakan untuk mengukur jarak dari 2 (dua) buah titik dalam *Euclidean space* (meliputi bidang euclidean dua dimensi, tiga dimensi, atau bahkan lebih). Jarak *Euclidean* digunakan untuk mencari jarak kedekatan antar citra template dan citra input dalam proses pengolahannya dengan KNN.

2.6 Python versi 3+



Gambar 2. 1 Logo Python

Penelitian ini akan dibangun dengan menggunakan bahasa pemrograman Python. Python merupakan bahasa pemrograman yang masih cukup baru dan populer dalam dunia pemrograman. Python sendiri sangat mendukung dalam pembuatan program penelitian ini yang berdasarkan pada *computer vision*.

Pembuatan aplikasi dalam penelitian ini menggunakan python dikarenakan kode yang mudah untuk dirancang bagi orang yang menggunakannya serta penggunaan spasi dalam mengidentasi kode yang sangat signifikan. Penambahan spasi pada kode tidak dapat dilakukan sembarangan, namun Python berifat portable dalam kode sumber. Kode sumber Python dapat digunakan di berbagai platform seperti

Windows, Mac Os, dan Linux, pernyataan ini diambil dalam Buku Logika Pemrograman Python oleh (Kadir, 2019).

Keunggulan python lainnya merupakan bahasa pemrograman yang mempunyai banyak kegunaan yang berfokus pada keterbacaan kode yang mampu menggabungkan sintaksis kode yang sangat jelas erta memiliki pustaka yang sangat besar, pernyataan ini disimpulkan yang diambil dalam Buku Semua Bisa Menjadi *Programmer Python Basic* oleh (Supardi 2017). Bahasa pemrograman python juga saat ini banyak digunakan dalam pembuatan program berbasis kecerdasan buatan terutama dalam *computer vision*.

2.7 OpenCV (*Open Source Computer Vision Library*)

Dalam pembuatan program dalam penelitian ini merujuk pada bidang *computer vision* sehingga menggunakan pustaka perangkat lunak berupa OpenCV. OpenCV sendiri mempunyai fungsi yang sangat tepat yaitu tersedianya pustaka yang dapat mengelolah citra penelitian ini. Menurut Jurnal Pseudocode oleh (Prabowo, Abdullah dan Manik, 2018) OpenCV adalah perpustakaan komputer open source visi dan perangkat lunak pembelajaran. OpenCV dibangun untuk menyediakan infrastruktur umum untuk aplikasi visi komputer dan untuk mempercepat penggunaan persepsi mesin dalam produk komersial. Kode *library* ini memiliki lebih dari 2500 algoritma yang dapat dioptimalkan, yang mencakup satu set lengkap visi komputer klasik dan *state-of-the-art* komputer vision dan mesin pembelajaran algoritma. OpenCV adalah metode yang paling cepat dan memiliki library paling lengkap untuk komputer vision.




Tidak hanya digunakan dalam bahasa pemrograman C saja, sekarang OpenCV juga bisa diinstall dalam Bahasa pemrograman python dengan mudah. Penggunaan OpenCV yang berguna sebagai *library* atau pustaka untuk memproses sebuah citra dalam kinerja *computer vision* sangatlah mendukung dan sesuai untuk diinstall pada python.

2.8 Diagram Alir (*Flowchart*)

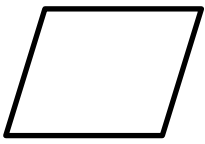
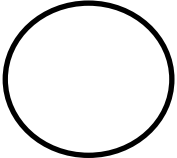
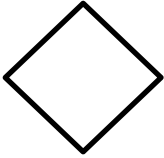
Penelitian ini dijalankan dengan membuat algoritma-algoritma singkat dan jelas yang akan dijabarkan menjadi proses berjalannya atau langkah-langkah program yang dibuat dalam penelitian ini agar dapat menghasilkan suatu alur proses yang dapat dipahami dalam logika dan nalar. Penggambaran algoritma-algoritma penelitian ini diwujudkan dalam Diagram alir (*Flowchart*).

Diagram alir (*flowchart*) adalah suatu diagram yang menggambarkan suatu proses untuk menggantikan algoritma maupun pseudocode. Diagram alir dibuat menggunakan beberapa simbol yang mempunyai fungsinya masing-masing. Berikut ini sejumlah simbol yang digunakan untuk menyusun diagram alir dalam Buku Logika Pemrograman Python oleh Kadir (2019):

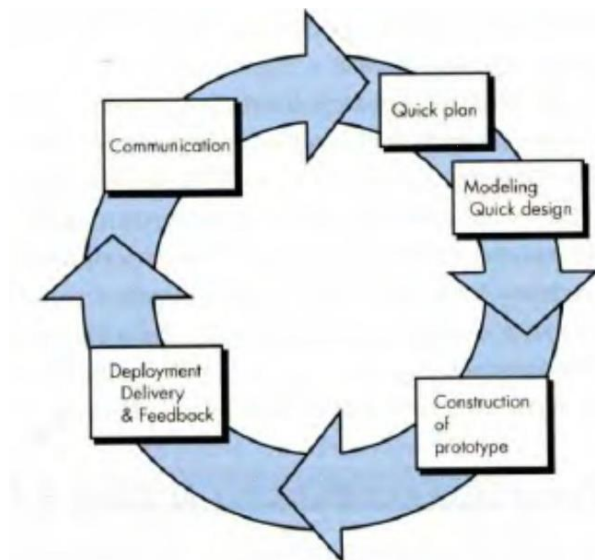
Tabel 2. 1 Bagan Alir Dokumen (*Document Flowchart*)

Simbol	Keterangan
Terminator 	Simbol ini digunakan untuk menyatakan titik awal atau titik akhir diagram alir.
Proses 	Simbol ini digunakan untuk menyatakan sembarang proses.
Proses Terdefinisi 	Simbol ini menyatakan prosedur lain yang telah didiagramalirkan pada tempat lain.

Tabel 2.1 (lanjutan)

Masukan / keluaran (data) 	Simbol ini menyatakan operasi masukan data atau penampilan data.
Konektor 	Simbol ini digunakan untuk menghubungkan ke berbagai bagian dalam diagram alir.
Keputusan 	Simbol ini digunakan untuk melakukan pengambilan keputusan.

2.9 Prototype Model



Gambar 2. 2 Diagram *Prototype Model*

Penelitian ini menggunakan metode pengembangan perangkat lunak *prototype*. Metode ini cocok untuk pengerjaan perangkat lunak yang hanya berjalan beberapa bulan saja dan mengurangi resiko kegagalan dalam proses penyelesaian masalah penelitian ini karena setiap pengerjaan tahapan selalu dilakukan analisa.

Menurut (Pressman, 2017) *Prototype* adalah proses pembuatan model sederhana *software* yang memungkinkan pengguna memiliki gambaran dasar tentang program serta melakukan pengujian awal. *Prototype* memberikan fasilitas bagi pengembang dan pengguna untuk saling berinteraksi selama proses pembuatan, sehingga pengembang dapat dengan mudah memodelkan perangkat lunak yang akan di buat. Metode ini cocok digunakan untuk mengembangkan sebuah perangkat lunak yang dikembangkan kembali.

Tahap-tahap pengembangan *Prototype model* menurut Pressman adalah :

1. *Communication* (komunikasi), pada tahapan ini, pengguna dan pengembang bersama-sama mengidentifikasi kebutuhan, format perangkat lunak, garis besar perangkat lunak yang akan dikembangkan.
2. *Quick Plan*, menggunakan hasil dari komunikasi dengan pengguna untuk membuat suatu tahapan perencanaan secara tepat.
3. *Modelling*, pada tahap pemodelan, termasuk didalamnya tahapan perancangan sederhana dilakukan untuk menggambarkan representasi dari perangkat lunak pada pengguna, misalnya interface atau format output dari perangkat lunak.
4. *Construction of Prototype*, setelah melakukan tahapan pemodelan, tahapan selanjutnya adalah pembuatan prototype dari perangkat lunak tersebut berdasarkan tahapan-tahapan sebelumnya.

5. *Deployment and Feedback*, prototype digunakan dan dievaluasi oleh pengguna perangkat lunak. Hasil dari evaluasi digunakan sebagai bahan masukan untuk proses pengembangan kembali dengan mengidentifikasi ulang kebutuhan perangkat lunak berdasarkan kekurangan dari prototype yang telah dihasilkan.

2.10 Penelitian Terkait

Berdasarkan masalah penelitian, penulis mengambil dan mendapat referensi dari beberapa penelitian yang berkaitan dan telah dilakukan sebelumnya. Berikut ini daftar penelitian terkait :

Tabel 2. 2 Penelitian Terkait

No	Nama	Judul	Tahun	Uraian
1.	Ikhwan Ruslianto dan Agus Harjoko	Pengenalan Karakter Plat Nomor Mobil Secara <i>Real Time</i>	2011	Penelitian ini dilakukan pada objek kendaraan beroda empat yang diambil secara real time dengan melakukan pengenalan karakter pada plat kendaraan, metode yang digunakan yaitu <i>connected component analysis</i> dan <i>template matching</i>
2.	Aris Budianto, Dwi Maryono dan Rosihan Ariyuana	Perbandingan <i>K-Nearest Neighbor</i> (KNN) Dan <i>Support Vector Machine</i> (SVM) Dalam Pengenalan Karakter Plat Kendaraan Bermotor	2018	Penelitian verifikasi kendaraan mengenai pengenalan karakter plat nomor kendaraan menggunakan metode Perbandingan <i>K-Nearest Neighbor</i> (KNN) dan <i>Support Vector Machine</i> (SVM) dengan menggunakan objek plat yang sudah <i>dicapture</i> .

Tabel 2.2 (lanjutan)

3.	Bernadus Bayu Prasetyo Aji , Suci Aulia dan Dadan Nur Ramadan	Perancangan Dan Realisasi Pengaman Sepeda Motor Berbasis <i>Face Recognition</i> Dengan <i>Realtime Database</i>	2019	Verifikasi kendaraan beroda dua dengan menggunakan pengenalan wajah sebagai system verifikasinya. Penelitian ini menggunakan metode template matching untuk menangani kepemilikan kendaraan berdasarkan wajah yang telah tersimpan pada <i>database</i> .
4.	Mohamad Dimiyati Ayatullaha, Devit Suwardiyanto dan I Wayan Suardinata	Implementasi Sidik Jari sebagai Otentikasi Parkir Kendaraan Menggunakan Raspberry Pi	2018	Penelitian yang dilakukan memanfaatkan identifikasi sidik jari pengemudi kendaraan dan pengenalan plat nomor kendaraan melalui foto oleh petugas parkir sebagai otentikasi. Sidik jari dideteksi dengan menggunakan sensor sidik jari yang kemudian disimpan dan saat kendaraan keluar sidik jari dicocokkan dengan mencari template yang sudah tersimpan.

