BAB 2
LANDASAN TEORI

2.1 Rasio pixel per metrik

Pixel Per Metrik adalah pengukuran yang digunakan untuk menentukan jumlah detail gambar potensial yang ditawarkan kamera pada jarak tertentu. Pengetahuan mengenai pixel per metrik ini digunakan untuk melakukan transformasi gambar dan pengenalan objek di dalam suatu gambar. Dalam perhitungan luas suatu objek dibutuhkan kalibrasi terhadap gambar yang digunakan [8].

Dalam melakukan kalibrasi untuk mengenali objek yang akan diukur digunakan objek referensi sebagai kalibrasi utama. Terdapat dua properti yang harus digunakan untuk mengenali suatu objek referensi dari suatu gambar yang digunakan. Pertama, ukuran (panjang atau lebar) dari suatu objek referensi harus diketahui terlebih dahulu. Kedua, pengidentifikasian suatu objek referensi berdasarkan penempatan yang telah ditentukan sebelumnya atau berdasarkan wujud baik warna atau bentuk dari objek referensi yang digunakan [9].

Setelah gambar dimuat, gambar akan ditransformasi untuk menemukan kontur objek. Setelah setiap kontur objek ditemukan, letak objek referensi yang telah ditentukan akan dihitung dari panjang kotak objek kontur yang dimuat (pixel) dan panjang asli objek referensi (centimeter) yang ditentukan menggunakan rumus berikut [6].

$$PpM = \frac{Lp}{Lo} \tag{2.1}$$

Pada rumus tersebut PpM berarti Pixel per metrik, L_p berarti panjang pixel suatu gambar dan L_o berarti nilai panjang observasi suatu benda. Disimpulkan dari rumus tersebut bahwa untuk setiap centimeter ditempati berarti panjang sebenarnya sama dengan setiap pixel yang dibagi.

2.2 Digital Image Processing

Beberapa langkah yang digunakan dalam pemrosesan gambar pada penelitian ini seperti: pengambilan gambar atau citra, pemrosesan warna gambar, pemrosesan morfologi, segmentasi, representasi dan pengenalan objek [10].

Berikut penjelasan dari masing-masing langkah.

1. **Pengambilan citra**: bertujuan untuk melakukan perubahan gambar optik analog menjadi gambar optik digital. Akuisisi citra digital berkaitan dengan pembentukan array 2D dari nilai integer yang mewakili refleksi pemandangan atau bentuk aktual pada interval khusus yang saling terpisah [11].

2. **Pemrosesan warna gambar**: bertujuan mengambil fitur yang terdapat pada citra atau gambar berdasarkan pendekatan warna. Pengubahan warna dilakukan pada penelitian menggunakan Pseudo-color image processing yang bertujuan untuk menetapkan warna ke nilai abu-abu berdasarkan kriteria yang ditentukan [12].

3. **Segmentasi**: merupakan proses melakukan partisi gambar menjadi banyak segmen. Prosedur ini memungkinkan untuk menemukan objek dalam gambar dan mengidentifikasi batas-batas objek [10]. Metode *discontinuty detection* adalah salah satu metode segmentasi gambar ke daerah berdasarkan diskontinuitas. Diskontinuitas yang didefinisikan adalah diskontinuitas tepi. Diskontinuitas tepi dihasilkan karena intensitas dikenali dan digunakan untuk menentukan batas area. Konsep penggunaan *discontinuity detection* adalah *contour detection* dan *histogram filtering* [13].

4. **Representasi dan deskripsi**: Representasi berhubungan dengan menunjukkan output gambar dalam wujud batas-batas dan/atau wilayah tertentu. Representasi dapat melibatkan karakteristik bentuk di sudut atau representasi regional seperti tekstur atau bentuk kerangka. Deskripsi berhubungan dengan ekstraksi informasi yang penting atau mencolok dari sebuah gambar. Informasi yang diekstraksi dapat membantu membedakan antara kelas objek satu sama lain secara akurat [10].

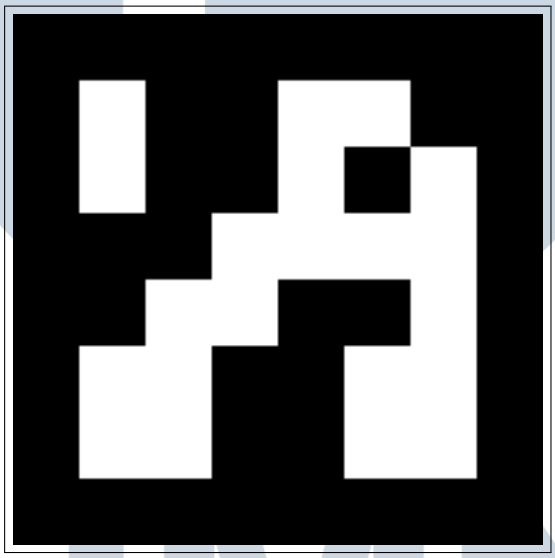
5. **Pengenalan objek**: merupakan proses pemberian label pada suatu objek tergantung pada deskripsinya untuk tujuan klasifikasi dari gambar yang dipindai [10].

2.3 ArUco Marker

ArUco Marker adalah penanda fidusia biner persegi yang dapat digunakan untuk estimasi pose kamera. Modul arUco di dalam OpenCV mencakup alat dan

fungsi pendeteksian penanda ini untuk digunakan dalam estimasi pose dan kalibrasi kamera. ArUco marker dinilai kuat, cepat dan sederhana untuk digunakan [14].

ArUco Marker terdiri dari grid biner tujuh kali tujuh. Setiap sel disebut bit. Kamus ArUco marker terdiri dari sejumlah penanda ArUco marker tersebut. Untuk mendeteksi penanda dalam sebuah foto, pertama-tama sudut *corner* penanda harus ditemukan, distorsi perspektif dikoreksi untuk mendapatkan gambar yang terlihat seolah-olah penanda dilihat dari atas, kemudian membagi gambar yang dihasilkan menjadi grid, dan membandingkan grid-grid bit putih dan hitam ini dengan kamus yang diberikan untuk mengetahui apakah ada kecocokan dari ArUco marker yang dipindai [15].



Gambar 2.1. ArUco Marker 6X6
Sumber: [14]

2.4 OpenCV

OpenCV adalah sebuah library open-source yang menyediakan kebutuhan untuk visi komputer, pemrosesan gambar dan pembelajaran mesin. OpenCV dapat digunakan untuk memproses gambar atau video untuk mengidentifikasi objek, pola, tulisan, wajah manusia dan lain sebagainya. Pendeteksian gambar dan berbagi fitur tersebut digunakan vector space dan perhitungan matematika untuk mengendalikan fitur-fitur tersebut [16]. Adapun fungsi-fungsi yang digunakan pada penelitian adalah sebagai berikut.

- cvtColor

Fungsi cvtColor digunakan untuk mengubah warna gambar [17].

- adaptiveTreshold
  Tresholding digunakan untuk menempatkan pixel ke dalam gambar yang berwarna abu-abu. Fungsi adaptiveTresholding adalah menetapkan pixel untuk gambar yang memiliki tingkat pencahayaan yang bermacam-macam agar mengurangi kesalahan dalam tresholding suatu area dalam gambar [17].
- findCountours
  Fungsi findContours digunakan untuk menemukan kontur yang membentuk suatu batasan objek dari suatu gambar [18].
- contourArea
  Fungsi contourArea digunakan untuk menghitung kontur dari suatu gambar [18].
- aruco.detectMarkers
  Fungsi aruco.detectMarkers adalah fungsi yang digunakan untuk mendeteksi keberadaan suatu objek ArUco marker di suatu gambar [19].
- aruco.DetectorParameters_create
  Fungsi aruco.DetectorParameters_create adalah bagian fungsi dari parameter yang berada pada aruco.detectMarkers. Fungsi ini digunakan membuat parameter tersebut saat menggunakan fungsi detectMarkers [19].
- aruco.Dictionary_get
  Fungsi aruco.Dictionary adalah fungsi yang digunakan untuk menerjemahkan ArUco marker yang dipindai dari suatu gambar [19].
- polylines
  Fungsi polylines digunakan untuk menggambar garis polyline pada suatu gambar [20].
- arcLength
  Fungsi arcLength digunakan untuk menghitung keliling kontur atau panjang kurva [18].
- minAreaRect
  Fungsi minAreaRect digunakan untuk menemukan persegi panjang yang diputar dari area minimum yang melingkupi set titik input gambar 2D. Fungsi

menghitung dan mengembalikan persegi panjang pembatas area minimum untuk suatu set titik yang ditentukan [18].

- boxPoints
  Fungsi boxPoints digunakan untuk menemukan 4 sudut, berfungsi untuk menggambar persergi panjang yang diputar [18].

- circle
  Fungsi circle digunakan untuk menggambar lingkaran pada suatu gambar[20].

- putText
  Fungsi putText digunakan untuk menggambar string text pada suatu gambar[20].

- imwrite
  Fungsi imwrite digunakan untuk menyimpan gambar ke suatu file yang ditentukan[21].

- imshow
  Fungsi imshow digunakan untuk menunjukkan gambar yang diminta[22].

- resize
  Fungsi resize digunakan untuk mengubah ukuran resolusi gambar sesuai yang diminta atau persentase ukuran yang diminta [17].

- moments
  Fungsi moments digunakan untuk menemukan titik tengah dan vertex dari sebuah objek kontur vektor 2d [18].

2.5 Root Mean Square Error (RMSE)

Root Mean Square Error adalah suatu cara perhitungan suatu tingkat kesalahan dari model dari suatu data kuantitatif. Semakin kecil nilai RMSE maka hasil pemindaian ukuran yang diterima akan semakin akurat. Nilai RMSE dapat dihitung menggunakan persamaan sebagai berikut[23]:

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(f-o)^2}{n}} \tag{2.2}$$

Pada persamaan berikut, f merupakan hasil yang dipindai. o merupakan hasil observasi atau nilai sesugguhnya. n merupakan jumlah data sampel yang dinilai.

2.6 Persentase Kesalahan (Approximation Error)

Persentase kesalahan adalah perbedaan antara nilai yang diukur atau eksperimen dan nilai yang diterima atau diketahui, dibagi dengan nilai yang diketahui, dikalikan 100%. Jika kesalahan persen mendekati nilai 0, maka nilai eksperimen aplikasi sangat dekat dengan nilai aktual atau benar. Untuk sebagian besar aplikasi, kesalahan persen direpresentasikan sebagai angka positif [24]. Rumus persentase kesalahan ditunjukan sebagai berikut:

$$\delta = \left| \frac{v_A - v_E}{v_E} \right| \cdot 100\% \tag{2.3}$$

Pada persamaan berikut, v_A merunjuk pada hasil observasi. v_E merunjuk pada hasil eksperimen.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA