

BAB II

TEORI PENUNJANG

Pada bab ini menjelaskan tentang teori rekonstruksi digital, Lidar dan beberapa komponen baik itu perangkat keras (*hardware*) dan juga perangkat lunak (*software*) yang perlu dipahami untuk digunakan dalam proses pembuatan alat dan sistem ini.

2.1 Computer Vision

Computer Vision merupakan ilmu yang mempelajari tentang bagaimana komputer dapat mengamati dan mengobservasi suatu benda tertentu. Hal ini bertujuan untuk meniru visualisasi yang dapat dilakukan oleh manusia sehingga dapat dilakukan oleh komputer, yang dalam aplikasinya dapat dilakukan pemrosesan lebih lanjut. *Computer Vision* terbagi menjadi beberapa bidang ilmu, diantaranya[5]:

1. Pengolahan Citra (*Image Processing*)

Pengolahan Citra merupakan salah satu ilmu yang mempelajari tentang teknik perbaikan kualitas citra yang kurang baik menjadi citra yang lebih baik dan diterima sebagai citra yang mudah dikenali.

2. Pengenalan Pola (*Pattern Recognition*)

Merupakan pengidentifikasian objek berdasarkan proses pengelompokkan data numerik dan simbolik (termasuk citra), yang dilakukan secara otomatis oleh komputer.

2.2 Rekonstruksi Digital

Rekonstruksi digital (juga disebut kartografi digital) adalah proses dimana suatu kumpulan data dikompilasi dan diformat ulang menjadi sebuah gambaran digital [6]. Teknologi ini memiliki fungsi utama yang berfungsi menghasilkan peta/gambaran yang memberikan representasi dari daerah tertentu, merinci jalan

utama dan tempat-tempat lainnya. Teknologi ini juga memungkinkan untuk perhitungan jarak dari suatu objek ke objek lain (bisa merupakan lokasi).

Dalam banyak kasus, pengguna dapat memilih antara gambaran digital, satelit (pandangan udara), dan pemandangan hybrid (kombinasi gambaran virtual dan pandangan udara). Dengan kemampuan dalam memperbarui dan memperluas perangkat prekonstruksi digital, jalan dan objek baru dapat ditampilkan pada gambar [7].

2.3 Simultaneous Localization and Mapping

Simultaneous Location and Mapping (SLAM) merupakan sebuah metode yang digunakan untuk pemetaan. Banyak digunakan pada pengembangan robot yang dapat melakukan pembacaan lingkungan dengan bantuan berbagai macam perangkat keras sebagai alat pengindraan, data yang didapat akan diolah menggunakan berbagai algoritma sehingga menampilkan hasil akhir berupa pemetaan keadaan sekitar.

Pengolahan data sensor Lidar pada teknologi SLAM menggunakan beberapa titik yang terbaca oleh sensor, dan menghitung titik-titik tersebut sehingga dapat menghasilkan citra keadaan dalam bentuk visual 2 dimensi atau 3 dimensi [8].

VSLAM (*Visual SLAM*) adalah sebutan yang merupakan ikatan penting antara CV dan robot/mesin. Ketika sebuah kamera digunakan baik sendiri atau dikombinasikan dengan sensor lain dan digunakan untuk sistem pemetaan, maka kondisi tersebut dinamakan VSLAM. *Autonomous robot* adalah contoh utama penggunaan metode ini, pernah digunakan pada *autonomous vehicle* [9], bahkan di dalam tubuh manusia [10].

2.3.1 Occupancy Grid Map

Occupancy grid digunakan untuk mewakili ruang kerja robot sebagai bagian terpisah. Informasi tentang lingkungan yang telah dikumpulkan dari sensor-sensor secara real time atau di ambil dari data percobaan-percobaan yang sebelumnya

pernah dilakukan. Kamera, sensor *ultrasound*, sensor kedalaman, pemindai laser biasanya dimanfaatkan untuk menemukan hambatan-hambatan atau objek-objek yang ada disekitar lingkungan robot.

Algoritma *occupancy grid map* digunakan dalam pembuatan robot agar mampu membuat perencanaan jalur. Algoritma tersebut digunakan dalam sistem pemetaan pada robot agar dapat mengintegrasikan informasi sensor dalam peta diskrit (*pixel*), dalam perencanaan lintasan untuk menemukan lintasan bebas hambatan, dan melakukan lokalisasi pada robot di lingkungan yang diketahui. Dalam pembuatan peta dapat dibuat dengan berbagai ukuran dan resolusi yang sesuai dengan sistem yang dibuat.

Dalam peta skala tipikal, untuk suatu titik, ia memiliki baik hambatan, atau tidak ada hambatan. Dalam pembuatan peta diskrit, untuk suatu titik dalam sel, digunakan $p(s = 1)$ untuk menunjukkan probabilitas bahwa ia berada dalam status bebas (*free*), $p(s = 0)$ untuk menunjukkan bahwa itu adalah probabilitas keadaan sel tersebut terdapat objek (*occupied*). Tentu saja, jumlah keduanya adalah 1.

$$Odd(s) \frac{p(s=1)}{p(s=0)} \quad (2.1)$$

Menurut formula Bayesian terhadap suatu pengukuran (z):

$$p(s = 1|z) \frac{p(z|s=1)p(s=1)}{p(z)} \quad (2.2)$$

$$p(s = 0|z) \frac{p(z|s=0)p(s=0)}{p(z)} \quad (2.3)$$

Untuk lebih simpel dengan menggunakan logaritma pada persamaan diatas rumusnya menjadi:

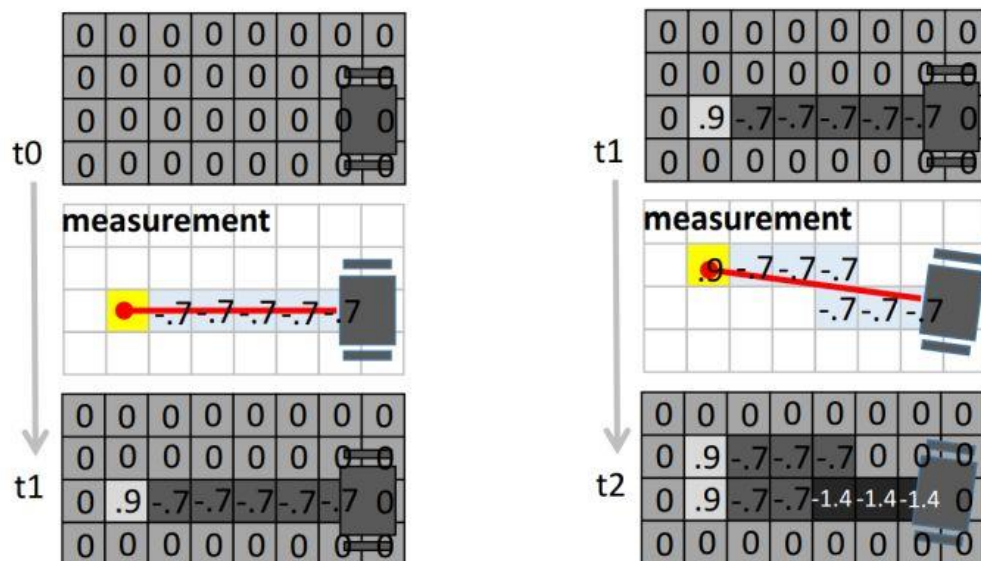
$$\log Odd(s|z) = \frac{p(z|s = 1)}{p(z|s = 0)} + \log Odd(s) \quad (2.4)$$

Lalu dari persamaan diatas, bisa kita pecah menjadi dua model perhitungan. Model perhitungan pertama untuk menentukan nilai parameter *lofree* dari titik-titik koordinat *free* dan nilai parameter *looccu* dari titik-titik koordinat *occupied*:

$$lofree = \log \frac{p(z = 0|s = 1)}{p(z = 0|s = 0)} \quad (2.5)$$

$$looccu = \log \frac{p(z = 1|s = 1)}{p(z = 1|s = 0)} \quad (2.6)$$

Setelah pemodelan seperti itu, Untuk memperbarui keadaan suatu titik, Anda hanya perlu melakukan penambahan dan pengurangan sederhana. Sebagai contoh, katakanlah kita atur *looccu* = 0,9, *lofree* = -0,7. Kemudian, semakin besar nilai parameter yang diberikan, semakin pasti itu adalah keadaan *occupied*. Sebaliknya, semakin kecil nilainya, semakin pasti keadaan bebasnya (Gambar II.1).



Gambar II.1 Proses data pemetaan [11].

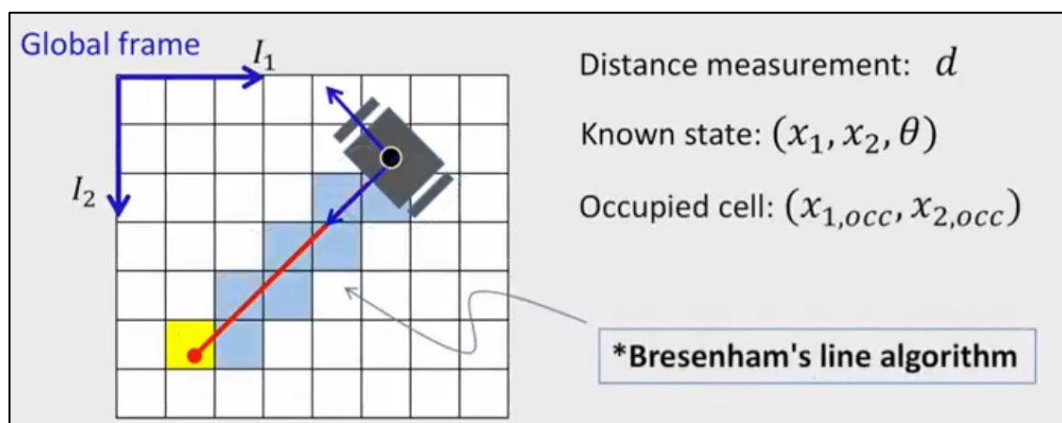
Gambar di atas menunjukkan proses memperbarui peta dengan data dari dua sensor laser. Dalam hasilnya, semakin gelap warna suatu titik, semakin pasti itu Gratis, dan semakin terang warnanya, semakin yakin itu ditempati.

2.3.2 Algoritma Bresenham

Algoritma bresenham adalah algoritma pembentukan garis yang digunakan untuk memperjelas kemungkinan titik *occupied* dan *free* pada masing-masing sel dari peta yang akan dibuat. Berdasarkan perhitungan umum dx maupun dy , menggunakan rumus:

$$dy = m \cdot dx \quad (2.7)$$

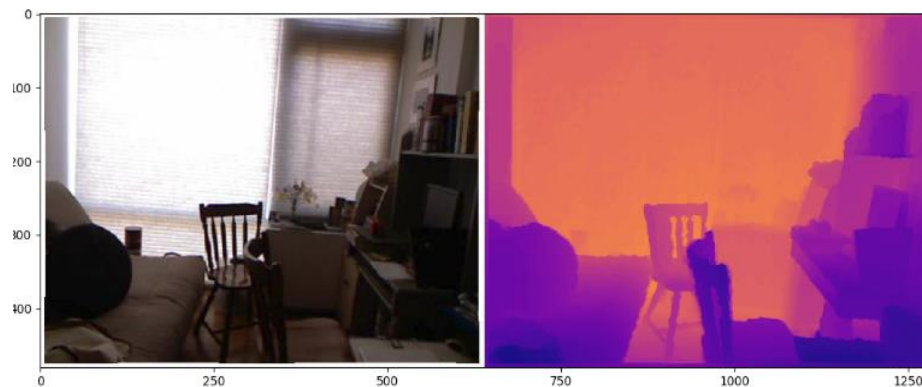
Pada dasarnya algoritma ini akan mengambil 2 titik koordinat (koordinat pemindai laser dan koordinat *occupied*) sebagai input bagi algoritma, dan menciptakan perkiraan garis lurus antara kedua titik koordinat. Sehingga dapat diketahui sel-sel diantara titik sensor dan titik objek adalah *free* (Gambar II.2).



Gambar II.2 Pemakaian algoritma Bresenham pada sistem.

2.4 *Depth Map*

Otak manusia memiliki kemampuan luar biasa untuk menyimpulkan kedalaman sebuah tampilan ketika melihat adegan 2D, bahkan hanya dengan satu titik pengukuran seperti dalam foto. Namun, pemetaan kedalaman yang akurat dari satu gambar tetap menjadi tantangan dalam bidang *Computer Vision (CV)*. Informasi kedalaman sebuah pemandangan sangat berharga untuk berbagai bidang lainnya, seperti *augmented reality*, robotika, *self-driving cars*, dan lain sebagainya. *Depth map* adalah sebuah metode yang memanfaatkan informasi dari gambar dengan membandingkan dengan jarak permukaan objek pemandangan dari sudut pandang tertentu, digunakan untuk memodelkan bentuk 3D atau merekonstruksinya (Gambar II.3).



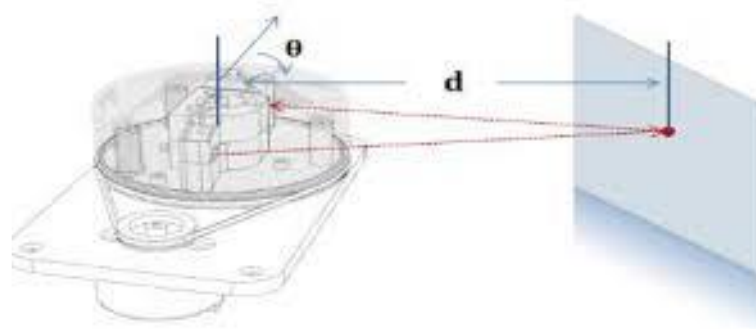
Gambar II.3 Contoh *Depth Map*.

2.5 *Light Detection and Ranging (Lidar)*

Light Detection and Ranging (Lidar) adalah sebuah teknologi sensor jarak jauh menggunakan properti cahaya yang tersebar untuk menemukan jarak dan informasi suatu objek dari target yang dituju. Metode untuk menentukan jarak suatu objek adalah dengan menggunakan pulsa laser [12].

Prinsip kerja Lidar secara umum adalah sensor memancarkan sinar laser terhadap objek kemudian sinar tersebut dipantulkan kembali ke sensor. Berkas sinar yang dipantulkan kemudian ditangkap dan di analisis oleh peralatan *detector*. Perubahan komposisi cahaya yang diterima dari sebuah target ditetapkan sebagai sebuah karakter objek.

Hasil dari proses Lidar adalah berupa sudut dan waktu perjalanan sinar saat dipancarkan dan diterima kembali diperlukan sebagai variabel penentu perhitungan jarak dari benda ke sensor[6]. Pada Gambar II.4 menunjukkan cara kerja dari sensor Lidar yang digunakan. Dari sensor Lidar ini akan didapatkan data keluaran berupa sudut dan jarak.

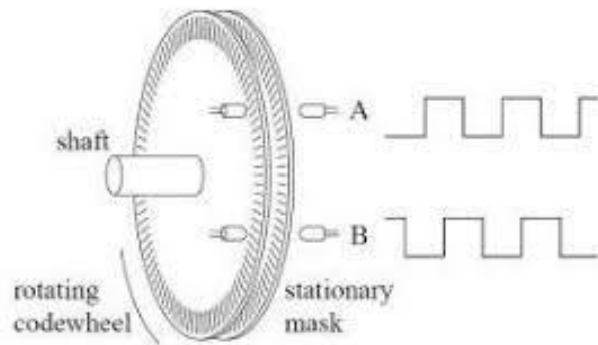


Gambar II.4 Prinsip Kerja Lidar [12].

Secara umum, data yang dihasilkan oleh Lidar disimpan dalam bentuk biner. Karena banyaknya data Lidar, memberikan dampak pada ukuran file yang dihasilkan. Jumlah data yang dihasilkan oleh Lidar akan meningkat sesuai dengan objek yang lebih kompleks [4].

2.5.1 Rotary Encoder

Rotary Encoder adalah perangkat elektromekanik yang dapat mendeteksi gerakan dan posisi (Gambar II.5).



Gambar II.5 Rotary Encoder [12].

Incremental encoder terdiri dari *double track* atau *single track* dan dua sensor yang disebut *channel A* dan *B*, hubungan fasa antara keduanya menghasilkan arah putaran. Dengan menghitung jumlah pulsa yang terjadi terhadap resolusi piringan maka putaran dapat diukur [12].

2.5.2 Time of Flight (ToF)

Prinsip ToF adalah jarak yang dapat diukur dengan memancarkan sinyal dengan kecepatan yang diketahui dan mengukur waktu yang dibutuhkan bagi sinyal yang dipantulkan ke sensor. Radar dan sonar juga bekerja dengan memanfaatkan konsep ini.

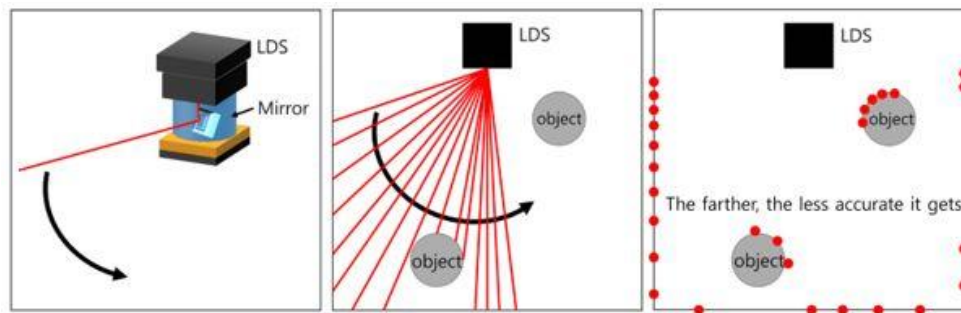
Secara historis hanya gelombang suara yang digunakan pada ToF dengan beberapa kelemahan, karena perambatannya yang lebih lambat dibandingkan gelombang lain sehingga sulit mengatasi pancaran sempit.

Untuk mengatasi masalah itu dapat menggunakan gelombang yang lebih tinggi, seperti: gelombang ultrasonic, cahaya, atau inframerah [4].

$$d = ct/2 \quad (2.8)$$

d adalah jarak yang dicari, c adalah kecepatan cahaya dan t adalah waktu yang didapatkan dari lama waktu sinyal kembali ke sensor.

Sensor Lidar akan mengirimkan paket dengan pola yang unik setiap melakukan transmisi, hal ini diperlukan agar sensor Lidar mudah mengidentifikasi gelombang tersebut. Konsep pembacaan Lidar dapat dilihat pada Gambar II.6.



Gambar II.6 ToF pada Lidar.

Setiap gelombang yang dikirimkan oleh Lidar masing-masing memiliki kecepatan yang spesifik, dan dapat terpengaruh oleh gravitasi. Waktu tempuh yang diperlukan oleh gelombang tersebut dari sensor kemudian mengenai sebuah obyek hingga kembali ke sensor dinamakan *time of flight* (ToF).

2.6 KITTI Dataset

KITTI adalah sebuah projek dari Institut Teknologi Karlsruhe dan Institut Teknologi Toyota yang berada di Chicago yang melakukan penelitian bersama di bidang *autonomous vehicle*. Berencana membuat *real-world computer vision*

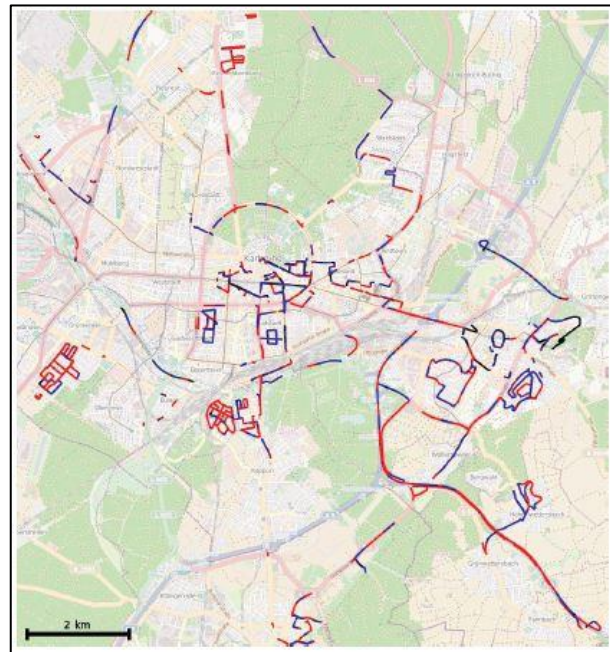
benchmark untuk *autonomous driving platform annieway*, dengan bidang minat: stereo, aliran optic, odometri visual, deteksi objek 3D dan pelacakan 3D [13].

Dataset KITTI telah direkam dari *platform* bergerak (Gambar II.7) saat mengemudi di dalam dan sekitar Karlsruhe, Jerman (Gambar II.8). Didalam dataset ini terdapat data gambar kamera, pemindaian laser, pengukuran GPS presisi tinggi dan akselerasi IMU dari sistem GPS / IMU gabungan. Tujuan utama dari pembuatan dataset secara *open source* ini adalah untuk membantu orang-orang yang ingin mempelajari dan mengembangkan sistem *autonomous vehicle* dengan memanfaatkan bidang ilmu *Computer Vision* [14].

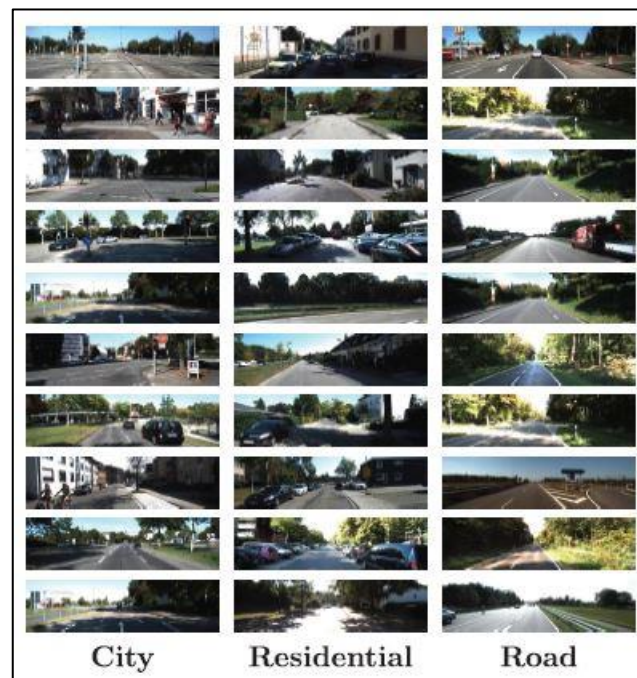
Pada data mentah yang disediakan berisikan beberapa kategori diantaranya ‘*Road*’, ‘*City*’, dan ‘*Residential*’. Contoh *frame* diilustrasikan pada Gambar II.9. Untuk setiap urutan, disediakan data mentah, anotasi objek dalam bentuk tracklet kotak pembatas 3D dan file kalibrasi, seperti yang diilustrasikan dalam Gambar II.10 [14].



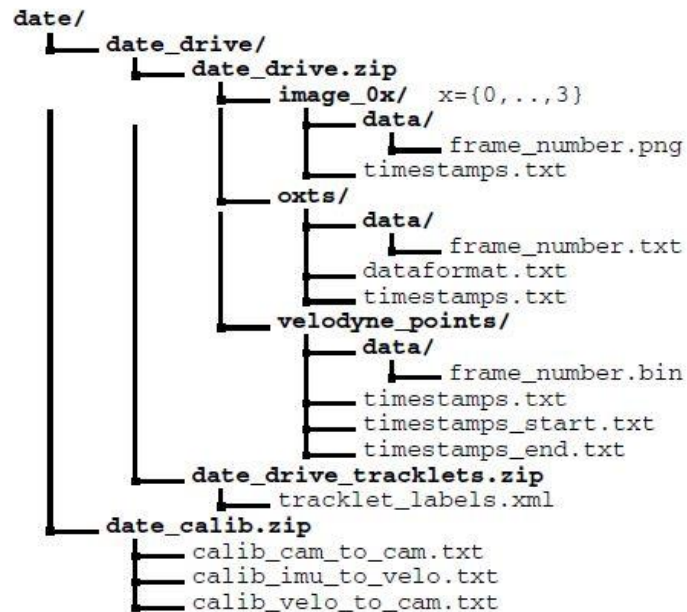
Gambar II.7 KITTI Autonomous Driving Platform Annieway [13].



Gambar II.8 Peta Karlsruhe, Jerman (Zona Perekaman) [14].



Gambar II.9 Kategori *scene frame* pengambilan gambar [14].



Gambar II.10 Struktur data mentah KITTI [14].

2.6.1 Folder Image

Pada folder ini berisikan foto yang diambil oleh kamera yang ada pada *autonomous car*. Foto yang diambil merupakan foto penampakan bagian depan mobil saja, dengan waktu pengambilan dan jumlah foto sesuai dengan pengambilan data lidar jadi dalam foldernya disediakan *timestamps* waktu dan tanggal pengambilan foto.

2.6.2 Folder OXTS

Pada folder ini berisikan data GPS (latitude, longitude, altitude) dan *car roll angle* yang menjelaskan posisi dari *autonomous car* pada peta yang sebenarnya. Pada sistem data OXTS ini akan dijadikan *pose* pada sistem untuk memperjelas orientasi dari *autonomous car*.

2.6.3 Folder Velodyne

Folder ini berisikan data yang diambil oleh sensor Lidar, berisikan sejumlah nomor/point file yang berekstensi .bin. Setiap point yang ada masing-masing telah tersimpan koordinat (x, y, z) dan nilai reflentansi (r), disediakan juga file-file yang berisikan waktu-waktu dari pengambilan data lidar sesuai dengan waktu pengambilan foto dari kamera.

2.6.4 Folder Calib

Folder ini berisikan file kalibrasi yang dapat digunakan pengembang dalam menstabilkan masing-masing data yang disediakan, mempermudah pengembang dalam membuat sistem-sistem rekonstruksi digital lingkungan.

2.7 Matlab

Matrix Laboratory (Matlab) adalah sebuah bahasa pemrograman yang cenderung di “kuasai” oleh bahasa yang telah mapan. Matlab hadir dengan fungsi dan karakteristik yang tidak sama dengan yang ditawarkan bahasa pemrograman lain (yang hampir sama biasanya). Matlab dikembangkan sebagai bahasa pemrograman untuk analisis dan komputasi *numeric* dan sekaligus alat visualisasi, yang menawarkan banyak kemampuan untuk menyelesaikan berbagai kasus, seperti bidang rekayasa Teknik, fisika, statistika, dan modeling.

Matlab dibangun dari bahasa induknya yaitu bahasa C, namun tidak dapat dikatakan sebagai varian dari C, karena dalam sintaks maupun cara kerjanya sama sekali berbeda dengan C. Namun, Matlab memiliki kelebihan-kelebihan bahasa C bahkan mampu berjalan pada semua platform Sistem Operasi tanpa mengalami perubahan sintak. Matlab memang dihadirkan bagi orang-orang yang tidak ingin disibukkan dengan rumitnya sintak dan alur logika pemrograman, sementara pada saat yang sama membutuhkan hasil komputasi dan visualisasi yang maksimal untuk mendukung pekerjaannya. [15].

Matlab juga berisi *toolbox* yang berisi fungsi-fungsi tambahan untuk aplikasi khusus. Kemampuan pemrograman yang dibutuhkan tidak terlalu sulit bila anda telah memiliki pengalaman dalam pemrograman bahasa lain seperti C, *PASCAL*, atau *FORTTRAN* [7]. Matlab merupakan merk *software* yang dikembangkan oleh *Mathwork.Inc* (<http://www.mathworks.com>) merupakan perangkat lunak yang paling efisien untuk perhitungan numerik berbasis matriks. Dengan demikian jika di dalam perhtungan kita dapat memformulasikan masalah ke dalam format matriks maka Matlab merupakan perangkat lunak terbaik untuk menyelesaikannya [16].

Beberapa fitur utama yang ada pada Matlab[7]:

1. Bahasa tingkat tinggi untuk penghitungan numerik.
2. Fungsi matematika untuk aljabar linier, statistic, analisis fourier, penyaringan (*filtering*), integrase numerik (*numerical integration*).
3. Lingkungan komputasi yang interaktif untuk eksplorasi dan pemecahan masalah.
4. *Built-in* grafis untuk memvisualisasikan data dan *tools* untuk membuat *custom plots*.

Sistem Matlab terdiri dari 5 bagian utama, yaitu [17]:

1. Bahasa (pemrograman) Matlab

Bagian ini adalah bahasa (pemrograman) tingkat tinggi yang menggunakan matriks atau array dengan pernyataan aliran kendali program, struktur data, masukan dan keluaran, serta fitur-fitur pemrograman berorientasi objek.

2. Lingkungan Kerja Matlab

Bagian ini adalah sekumpulan kakas dan fasilitas Matlab yang digunakan oleh pengguna atau pemogram.

3. Penampangan Grafik

Bagian ini adalah sistem grafik Matlab, termasuk perintah-perintah (program) tingkat tinggi untuk visualisasi data 2D dan 3D, pengolahan citra, animasi, dan presentasi grafik.

Selain itu bagian ini juga termasuk perintah-perintah (program) tingkat rendah untuk menetapkan sendiri tampilan grafik seperti halnya membuat antarmuka pengguna grafis untuk aplikasi-aplikasi Matlab.

4. Pustaka (library) fungsi matematis Matlab

Bagian ini adalah koleksi algoritma komputasi mulai dari fungsi dasar seperti menjumlahkan (sum), menentukan nilai sinus, kosinus, dan aritmatika bilangan kompleks, fungsi-fungsi seperti inverse matriks, nilai eigen matriks, fungsi Bessel, dan FFT (Fast Fourier Transform).

5. API (*Application Program Interface*) Matlab

Bagian ini adalah pustaka (library) untuk menuliskan program bahasa C dan Fortran berinteraksi dengan Matlab, termasuk fasilitas untuk memanggil rutin program dari Matlab (*Dynamic Linking*), memanggil Matlab sebagai mesin komputasi dan untuk pembacaan serta penulisan *MAT-Files*.