

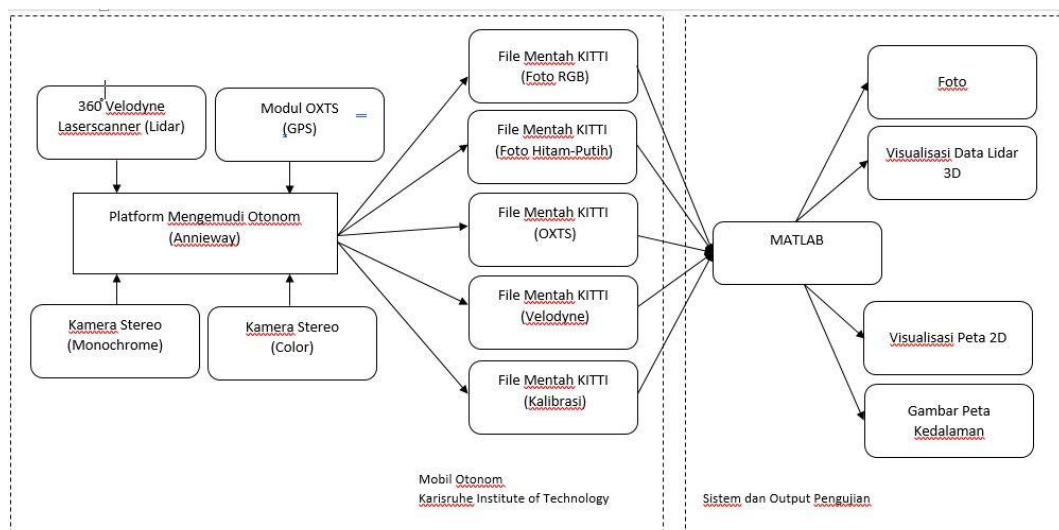
BAB III

PERANCANGAN SISTEM

Pada Bab ini akan membahas seluruh perancangan sistem yang akan dibuat. Perancangan sistem ini menggunakan bahasa pemrograman Matlab.

3.1 Diagram Blok

Perancangan diagram blok meliputi diagram blok secara keseluruhan dan fungsi-fungsi dari setiap blok. Diagram blok sistem secara keseluruhan dapat dilihat pada Gambar III.1.



Gambar III.1 Diagram Blok Secara Keseluruhan.

Pada sistem yang akan dibuat membutuhkan beberapa *dataset* dari KITTI yaitu data kalibrasi, Lidar, OXTS (GPS) dan foto/gambar sebagai acuan pemetaan dari lingkungan yang dilewati oleh *autonomous car*. Sebelumnya kita harus mendownload *dataset* yang dibutuhkan dari Velodyne *Laserscanner* (Lidar), GPS, dan kamera (dibebaskan memilih salah satu antara gambar berwarna dan abu-abu)

dari *website* (http://www.cvlibs.net/datasets/kitti/raw_data.php). Setelah itu data-data tersebut harus dikelompokkan dalam satu folder untuk memudahkan sistem manajemen pengambilan data untuk di proses.

Setelah sistem pada matlab mampu mengambil data maka akan diproses data Lidar terlebih dahulu untuk dapat menentukan batas titik-titik pantulan dari Lidar yang terkena objek. Dikarenakan data Lidar itu berupa bilangan biner *floating point* maka sistem harus mengubahnya menjadi data *integer* dengan konversi big-endian machine sehingga menghasilkan titik koordinat x, y, z, dan nilai reflektasi tambahan (yang bisa dihapus).

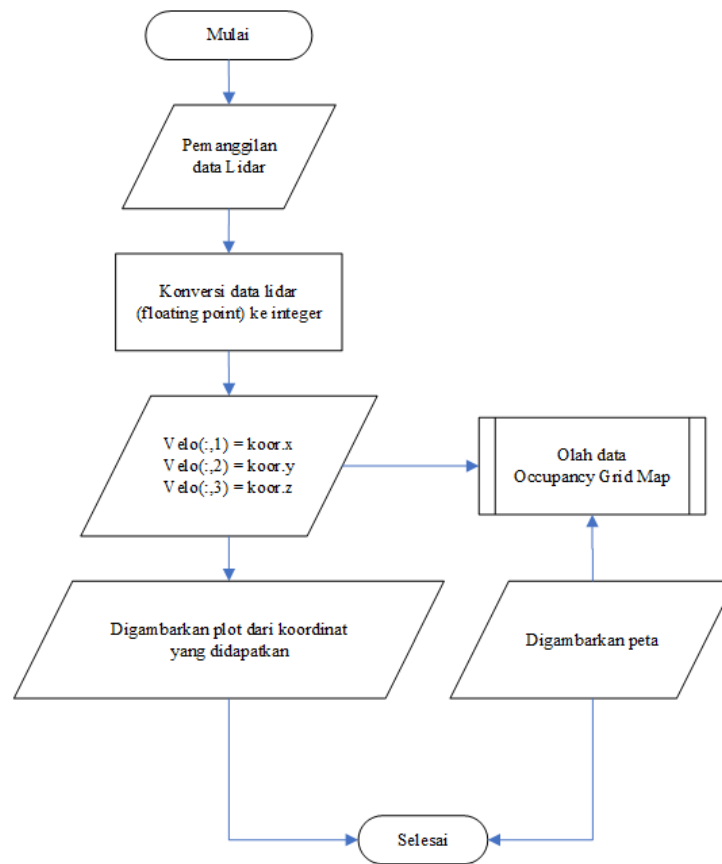
Setelah mendapatkan titik koordinat maka ditampilkanlah sebuah gambaran plot dari data Lidar yang merekonstruksi lingkungannya yang sebenarnya. Setelah itu sistem baru dapat mengintegrasikannya dengan gambar/foto yang diambil oleh kamera untuk memastikan apakah data LIDAR sudah benar mereplikasi lingkungannya dan dapat mengukur jarak permukaan objek yang ada di sekeliling mobil.

3.2 Perancangan Perangkat Lunak

Perancangan ini bertujuan untuk membuat sebuah algoritma agar sistem yang dibuat dapat beroperasi sesuai dengan tujuan pada penelitian ini. Algoritma program yang dirancang ini akan ditanam dan ditulis menggunakan bahasa pemrograman Matlab yang merupakan pemrograman terstruktur. Maka dari itu, algoritma perancangan program yang mengeksekusi data lidar ini digambarkan dalam bentuk *flowchart* (diagram alir).

3.2.1 Diagram Alir Data Lidar pada Sistem

Perancangan diagram alir pemrosesan data Lidar ini untuk menggambarkan bagaimana proses pengubahan data Lidar menjadi sebuah tampilan gambar (plot) dengan memanfaatkan metode SLAM, ditunjukkan pada Gambar III.2.



Gambar III.2 Diagram Alir Data Lidar.

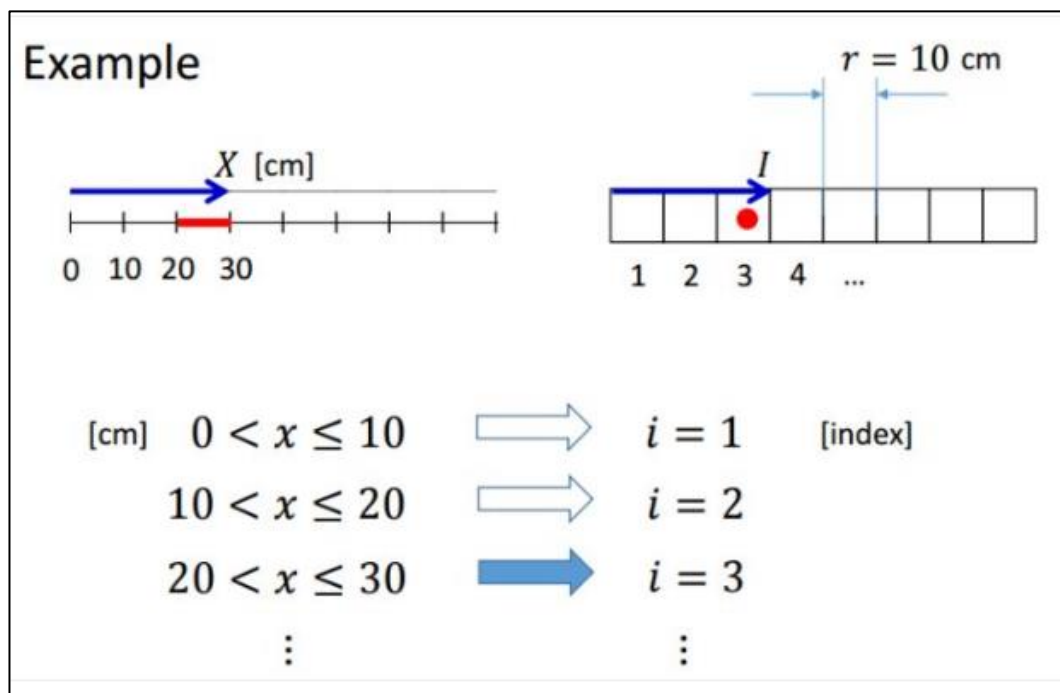
Dari data Lidar yang sudah dikonversi menjadi koordinat dalam bentuk integer, maka langkah selanjutnya menampilkan koordinat yang didapat pada sebuah plot dan melakukan pengolahan koordinat menggunakan algoritma *occupancy grid map*, agar dapat digambarkan pada sebuah peta.

3.2.2 Analisis Program Pemetaan

Pada program pemetaan akan digunakan beberapa input agar dapat diolah menjadi sebuah peta. Diantaranya, *local_occs* adalah *array* dari hasil scan data sensor Lidar; *pose_robot* adalah *array* dari data GPS (OXTS) yang berisikan informasi posisi dan orientasi mobil pada saat pengambilan data; *T_IMU2velo* adalah kalibrasi GPS dan data Lidar; *param* adalah beberapa parameter yang

digunakan, *param.resol* merupakan nilai resolusi peta (jumlah sel yang dibuat mewakili 1 meter pada peta dunia nyata), *param.origin* merupakan titik awal mobil pada sel di peta, *param.lo_occ* dan *param.lo_free* masing-masing adalah nilai parameter dari *looccu* dan *lofree*, *param.lo_max* dan *param.lo_min* adalah ambang status posisi.

Untuk menandakan koordinat yang didapat dari data Lidar pada peta (dalam ukuran pixel) yang akan dibuat maka kita harus mengubah koordinat di dunia nyata ke koordinat pada sel (Gambar III.3).



Gambar III.3 Contoh pengubahan skala.

Pada gambar X merupakan koordinat di dunia nyata dan I koordinat di peta, bisa kita lihat r adalah ukuran skala perbandingan (1 sel pada peta merupakan 10 cm pada dunia nyata). Sehingga pehitungannya menjadi

$$I = \text{sel}(x/r) \quad (3.1)$$

Jika kita memiliki 2 koordinat maka koordinat di peta, maka perhitungannya:

$$(I, J) = \left(\text{sel} \left(\frac{X}{r} \right), \text{sel} \left(\frac{Y}{r} \right) \right) \quad (3.2)$$

Setelah kita mengetahui koordinat yang digunakan (free dan occupied) dan dapat mengubahnya ke koordinat pada peta, berikut merupakan program *function* dari algoritma *occupancy grid map* dan penggunaan *private function* dari algoritma bresenham.

```
function myMap=occGridMapping(myMap,local_occs,pose_robot,param,T_IMU2velo)
% % the number of grids for 1 meter.
resol = param.resol;
% % the origin of the map in pixels
origin = param.origin;
lo_occ = param.lo_occ;
lo_free = param.lo_free;
lo_max = param.lo_max;
lo_min = param.lo_min;

lidarn = size(local_occs,1); % number of rays per timestamp
pose_lidar= T_IMU2velo*pose_robot;

% coordinate of robot in real world
x = pose_lidar(1,4);
y = pose_lidar(2,4);
% coordinate of robot in metric map
grid_rob = ceil(resol * [x; y]);

% calc coordinates of occupied and free points in metric map
for j=1:lidarn
    real_occ = pose_lidar * [local_occs(j,:)';1];
    grid_occ = ceil(resol * real_occ(1:2,:)); % coordinate of occ in metric map
    % coordinates of free in metric map (by bresenham's algorithm)
    [freex, freey] = bresenham(grid_rob(1),grid_rob(2),grid_occ(1),grid_occ(2));
    % convert coordinate to offset to array
    free = sub2ind(size(myMap),freey+origin(2),freex+origin(1));
    occ = sub2ind(size(myMap), grid_occ(2)+origin(2), grid_occ(1)+origin(1));
    % update metric map
    myMap(free) = myMap(free) - lo_free;
    myMap(occ) = myMap(occ) + lo_occ;
end
myMap(myMap < lo_min) = lo_min;
myMap(myMap > lo_max) = lo_max;
```

Gambar III.4 Program Matlab (sistem *mapping*).

3.3 Implementasi Sistem

Setelah dilakukan perancangan dan analisis terhadap perangkat lunak yang akan dibangun maka langkah selanjutnya adalah mengimplementasikan hasil-hasil tersebut [18]. Perangkat lunak ini dibangun dan diuji menggunakan perangkat komputer dengan spesifikasi sebagai berikut:

1. Processor intel core i3 berkecepatan 3.5 GHz.
2. RAM 4 GB.
3. *Harddisk* berkapasitas 500 GB.
4. *VGA card* NVIDIA GeForce 210.
5. Monitor LCD 18.5” dengan resolusi 1366x768 pixel.
6. *Mouse* dan *keyboard*.