BAB II

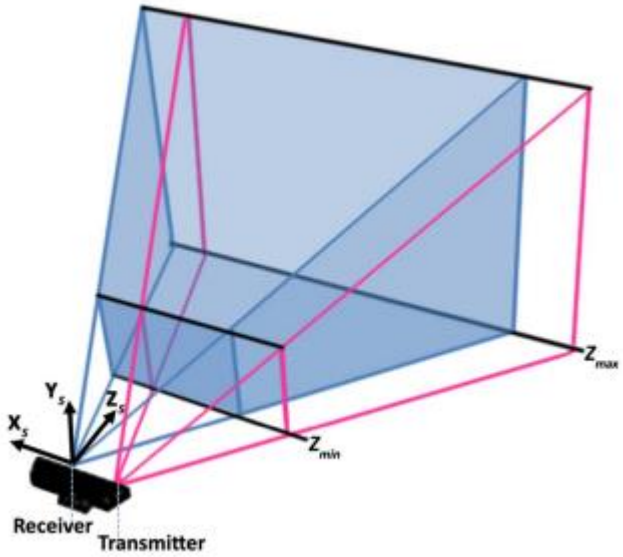
TINJAUAN PUSTAKA

2.1 Persepsi Visual

Persepsi visual merupakan fungsi fundamental untuk menjalankan sebuah robot, persepsi lingkungan ini menyediakan informasi penting bagi robot untuk beroperasi pada suatu lingkunga, termasuk lingkungan yang dapat dilewatin, objek yang berada disekitar lokasi, kedalaman atau jarak dan bahkan prediksi akan masa depan yang akan terjadi pada lingkungan tersebut seperti prediksi suatu barang akan jatuh dan sebagainya [16]. Berdasarkan dari sensor yang diimplementasikan, tugas persepsi lingkungan ini dapat ditangani menggunakan berbagai sensor seperti *ultrasonic*, lidar, sensor inframerah, kamera RGB atau perpaduan antara beberapa jenis perangkat ini, namun pada penelitian ini, penulis akan menggunakan sensor inframerah dan kamera RGB dikarenakan beberapa fitur yang unggul dibandingkan sensor ultrasonic dan lidar seperti yang telah dipaparkan pada pendahuluan, dua elemen persepsi lingkungan yang akan dicapai setelah sensor akan diimplementasikan ialah mendeteksi kedalaman untuk memperoleh jarak, lalu mendeteksi objek pada lingkungan robot berada.

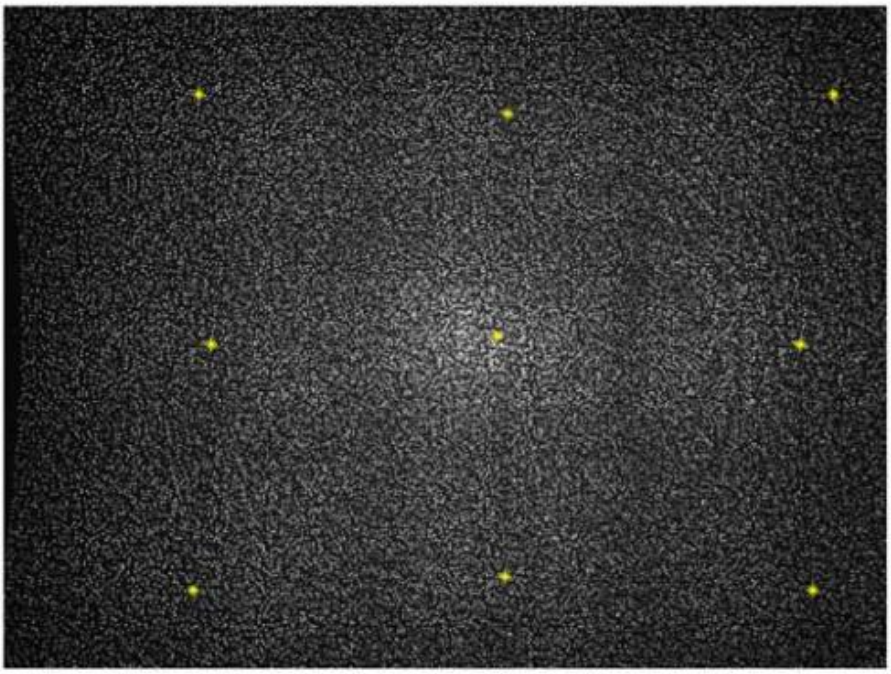
2.2 Sensor Jarak Berbasis Inframerah

Microsoft kinect telah menjadi sensor inframerah yang sangat populer dan banyak dipakai untuk tujuan penelitian. Dengan total penjualan melebihi 24 juta unit, ada juga beberapa komunitas developer yang membagi *resources* yang melibatkan penggunaan *kinect*, seperti pengimplementasian *kinect* pada robot dan *microsoft developer network*, dan hampir semua kelompok penelitian mem aplikasikan sensor *kinect* ini dikarenakan memiliki sensor kedalaman yang akurat untuk beberapa aplikasi [5]. Peta kedalaman sendiri merupakan istilah yang diberikan pada hasil pencitraan baik menggunakan kamera stereo maupun laser, penggunaan kata kedalaman ini merupakan penjelasan dari jarak objek pada citra dengan titik perekaman sensor [5].



Gambar 2.1 Sensor inframerah memancarkan 1 [5]

Sistem Kinect menggunakan 2 sensor untuk mengkonstruksikan gambar kedalaman. Yaitu sensor 1M IR Laser yaitu pemancar sinar laser inframerah dan Aptina MT9M001 CMOS sensor, yang kemudian disejajarkan untuk memiliki sumbu optik parerel dan diimbangi oleh jarak dasar antar kedua sensor ialah 75 mm [5]. Proyektor laser dan sensor gambar IR monokrom keduanya bekerja dalam stereo sebagai sistem trangulasi aktif matrik dimana titik pola IR ditransmisikan dan diterima.



Gambar 2.2 Sinar inframerah kinect yang telah dipancarkan

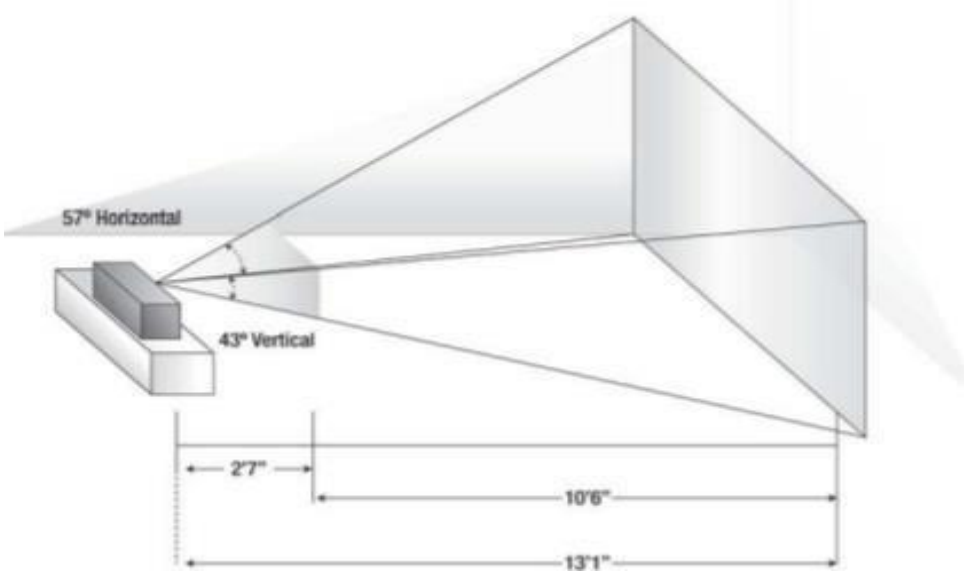
Pada mulanya Kinect adalah alat pendeteksi gerakan dan suara untuk digunakan pada konsol permainan Xbox 360 dan Xbox One yang diproduksi oleh Microsoft [8]. Pada tanggal 16 Juni 2011, Microsoft merilis *Kinect Software Development Kit* (Kinect SDK) yang dapat digunakan oleh para pengembang piranti lunak untuk memgembangkan aplikasi yang menggunakan Kinect. Sejak saat itu, Kinect banyak digunakan dalam penelitian computer vision. Kinect memiliki beberapa fitur utama yang meliputi:

1. Raw sensor stream yang dapat digunakan untuk mengakses informasi dari *depth sensor, color camera sensor, dan microphone array.*
2. *Skeletal tracking* yang dapat digunakan untuk mendeteksi skeleton dari pengguna yang berada dalam jangkauan jarak pandang Kinect.
3. *Advanced audio capabilities* yang memiliki beberapa kegunaan utama seperti *effective noise suppression, acoustic echo cancellation, beamforming, dan source locatization*

Gambar 2. 3 Sensor Inframerah 1M IR Laser dan Aptina MT9M001 CMOS pada Kinect XBOX 360 [4]

Kinect memiliki batasan-batasan penggunaan tertentu agar penggunaan sensor kamera yang ada padanya dapat bekerja secara optimum. Batasan- batasan tersebut adalah:

1. Sudut pandang horizontal sebesar 57 derajat. 2.Sudut pandang vertikal sebesar 43 derajat.
2. Berada pada suhu ruangan 50 sampai dengan 350 derajat Fahrenheit.
3. Jarak pengguna yang berada pada 1,2 sampai dengan 4 meter untuk standard mode (0,4 sampai dengan 3 meter untuk *near mode*).
4. Jangkauan depth yang dihasilkan berada pada jangkauan 400 mm untuk near mode sampai 8000 mm untuk *standard mode*.

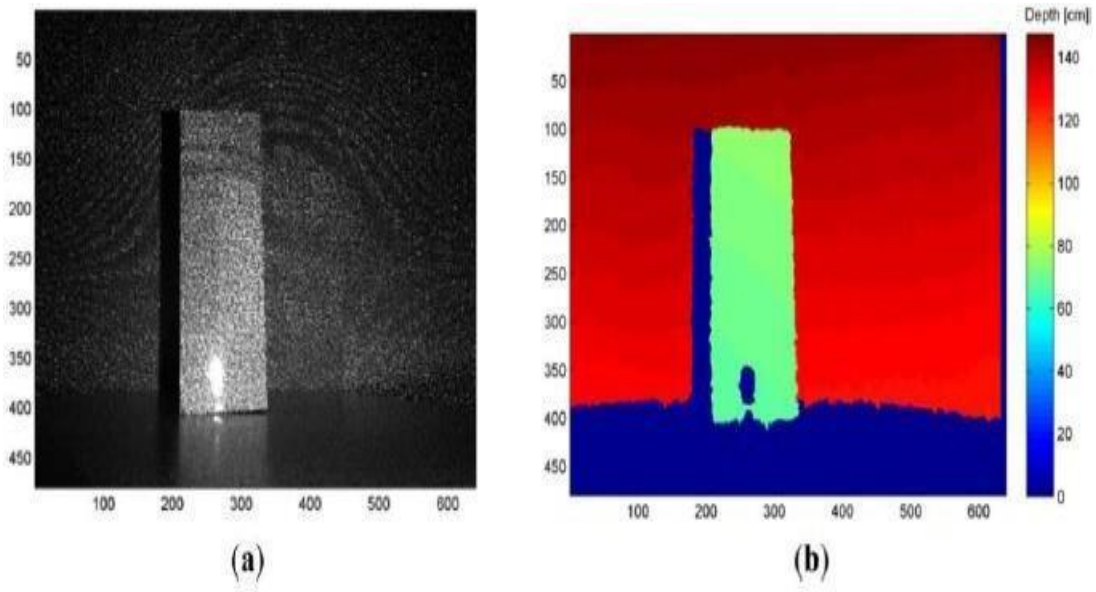


Gambar 2. 4 Jangkauan jarak pandang Kinect [4]

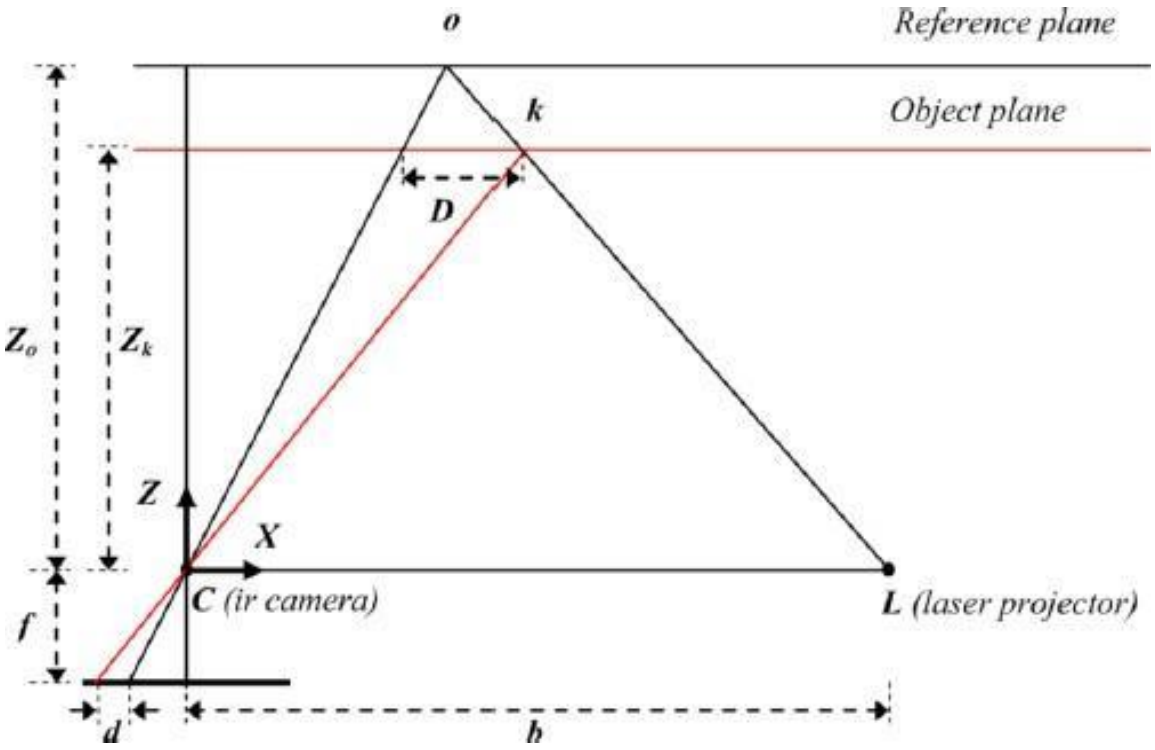
2.2.1 Pengukuran Geomteri Kinect

Sensor pada Kinect terdiri dari emitor laser inframerah, kamera inframerah dan kamera RGB. Pengukuran kedalaman (depth) pada Kinect merupakan proses triangulasi. Sumber laser memancarkan sinar tunggal yang dibagi menjadi beberapa sinar oleh kisi difraksi untuk menciptakan pola yang konstan dari suatu objek yang diproyeksikan dalam sebuah bidang. Pola tersebut ditangkap oleh kamera inframerah dan berkorelasi dengan pola referensi. Pola referensi diperoleh dengan menangkap objek pada jarak yang telah diketahui oleh sensor dan disimpan dalam memori sensor. Ketika objek diproyeksikan pada objek yang jarak kesensornya lebih kecil atau lebih besar dari referensi, posisi dari objek pada gambar inframerah akan bergeser ke arah baseline antara proyektor laser dan pusat perspektif kamera inframerah [9]. Pergeseran ini diukur untuk semua objek

dengan korelasi gambar sederhana yang akan menghasilkan sebuah gambar disparitas. Gambar 2.3 menggambarkan pengukuran kedalaman dari pola objek.



Gambar 2. 5 Pengukuran kedalaman pada objek [9]



Gambar 2.6 Hubungan antara kedalaman (depth) realtive dan pengukuran dispartis(perbedaan) [9]

Dimana :

o = Proyeksi objek pada bidang referensi k = Titik objek

d = Perbedaan pengamatan

b = Panjang dasar (*baseline*)

D = Perpindahan dari titik k dalam ruang objek f= Panjang fokus kamera inframerah

Zo = Jarak kamera terhadap proyeksi objek pada bidang referensi

Zk = (*depth*) dari titik k dalam ruang objek X= Sumbu X kamera

Z = Sumbu Z kamera

Pada Gambar 2.4 dapat dilihat hubungan antara jarak titik objek k ke sensor relatif terhadap bidang referensi dan diukur besar disparitas d. Untuk mendapatkan koordinat 3D dari titik objek perlu mempertimbangkan origin sistem koordinat depth dipusat perspektif kamera inframerah. Sumbu Z orthogonal terhadap bidang gambar ke arah objek, sumbu X tegak lurus terhadap sumbu Z kearah baseline b antara pusat kamera inframerah dan proyektor laser, dan sumbu Y orthogonal terhadap sumbu X dan sumbu Z mengikuti kaedah sistem koordinat tangan kanan. asumsikan bahwa sebuah objek pada bidang referensi pada jarak Zo ke sensor, dan Speckle pada objek yang di rekam pada bidang gambar dari kamera inframerah. Jika objek k digeser mendekati atau menjauhi sensor, lokasi dari Speckle pada bidang gambar akan dipindahkan kearah X. Hal ini diukur dalam ruang gambar sebagai perbedaan sesuai dengan titik k dalam ruang objek. Dengan demikian berdasarkan konsep segitiga sebangun, maka didapat persamaan 2.1 dan persamaan 2.2 berikut

$\frac{D}{b} = \frac{Z_0 - Z_k}{Z_0}$ ……**Persamaan 2.1**

$\frac{d}{f} = \frac{D}{Z_k}$ ……...**Persamaan 2.2**

Zk = kedalaman (depth) dari titik k dalam ruang objek,

b = panjang dasar (baseline),

f = panjang fokus kamera inframerah,

D = perpindahan dari titik k dalam ruang objek,

d = perbedaan yang diamati dalam ruang gambar.

Substitusikan D pada persamaan 1 ke persamaan 2 maka akan diperoleh nilai Zk berdasarkan dari variable lainnya, yaitu dengan persamaan 2.3 berikut:

dimana:

$Z_k = \frac{Z_0}{1 + \frac{Z_0}{fb} d}$ ……..**Persamaan 2.3**

Zk = (*depth*) dari titik k dalam ruang objek ,

Zo= Jarak kamera terhadap proyeksi objek pada bidang referensi,

d = perbedaan yang diamati dalam ruang gambar,

b = panjang dasar (*baseline*),

f = panjang fokus kamera inframerah.

Persamaan 3 merupakan dasar model matematis untuk derivasi kedalaman (*depth*) dari perbedaan yang diamati, asalkan parameter konstan Zo, f, dan b dapat ditentukan dengan kalibrasi. Berdasarkan nilai Z dan f dapat mendefinisikan skala pencitraan pada saat itu, sehingga koordinat planimetris X dan Y dari setiap titik dapat dihitung dari koordinat citra dan skala dengan persamaan 2.4 dan persamaan 2.5 sebagai berikut :

$X_k = -\frac{Z_k}{f}(x_k - x_0 - \delta x)$……...**Persamaan 2.4**

$Y_k = -\frac{Z_k}{f}(y_k - y_0 - \delta y)$…….....**Persamaan 2.5**

Dimana:

xk, yk = koordinat citra dari titik k,

$x0$, $y0$ = koordinat titik utama (*principle point*),

δx, δy = koreksi distorsi lensa.

2.4 Sistem Deteksi Dan Pengenalan Objek *Berbasis Convolutional Neural Network*

YOLO merupakan perwakilan dari metode deteksi satu tahap tanpa tahap region proposal yang berbeda, yang melakukan tugas deteksi objek sebagai masalah regresi tunggal. Gagasan intinya ialah untuk memakai seluruh citra sebagai input kedalam *network* dan langsung dari piksel gambar ke koordinat kotak pembatas *atau bounding box* dan *class probabilitas*. Peta fitur lapisan *output* dari YOLO dirancang untuk memprediksi koordinat *bounding box*, *confidence* pendeteksian, dan *probablilitas class*, dengan demikian YOLO memungkinkan deteksi beberapa objek dalam jaringan neural. Oleh karena itu, kecepatan mendeteksi lebih cepat daripada metode konvensional. Namun karena keterbatasan grid 7 x 7 kesalahan lokalisasi besar dan akurasi deteksi rendah. YOLO tidak mendeteksi objek kecil baik benda benda kecil terutama padat, dikarenakan satu grid hanya bisa memprediksi 1 objek. Untuk mengatasi masalah ini, YOLOv2 telah diusulkan, YOLOV2 meningkat akurasi deteksi melalui langkah berikut ini:

1. *Batch normalization* (BN), dengan menambahkan *Batch Normalization* pada semua layer convulutional dapat meningkatkan akurasi dalam mendeteksi.
2. *High resolution classifier,* ukuran awal dari citra input sebesar 224 x 224 digantikan

dengan ukuran 448 x 448 untuk training jaringan class, High resolution membuat pendeteksian bekerja lebih bagus.

3. *Convolutional with anchors boxes*, pada original YOLO lapisan lapisan atau layer yang saling terhubung dihapus dan diganti dengan *anchors boxes* untuk mendeteksi *bounding box*. Adopsi *anchors boxe*s ini membuat sedikit penurunan pada akurasi tetapi meningkatkan ingatan dan peluang untuk mendeteksi semua objek.
4. *Dimension clusters*, pada *tarining set bounding box*, metode pengelompokan *Kmeans* digunakan untuk secara otomatis untuk menemukan prioritas yang baik. Untuk mengurangi kesalahan yang disebabkan oleh ukuran kotak yang berbeda, skor IOU *(Intersection Over Union*) digunakan untuk pengelompokan.
5. *Direct location predction. Predicting* koordinasi lokasi yang berelasi dengan lokasi *grid cell* dibandingkan dengan *offset* pada anchors yang mana membuat jaringan (*network*) lebih stabil dan mudah untuk dipelajarin.
6. *Fine-grained features*. Untuk memimprove kemampuan dalam mendeteksi objek kecil, YOLOV2 menambahkan fungsi *pass-through kayer*, yang mengkombinasikan fitur *high resolution* dengan fitur *low resolution.*
7. *Multi-scale training. Multi-scale training* ini membantu jaringan untuk mempelajari bagaimana untuk memprediksilintas dimensi input, ini berarti pendeteksian dengan resolusi berbeda dapat diprediksi menggunakan jaringan yang sama.

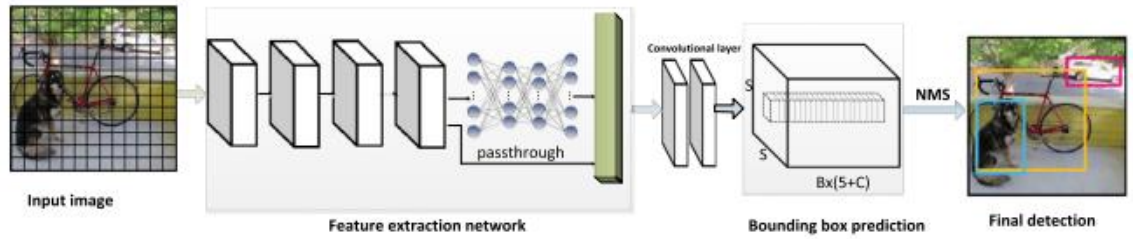
Bagaimanapun, akurasi deteksi pada YOLOV2 masih kecil pada objek kecil. Karena itu YOLOV3 telah membuat beberapa improvisasi pada basis YOLOV2.

1) Klassifikasi multi-label pada YOLOV3 menggunakan pengklasifikasi logistik indenpenden bukan softmax classifier untuk prediksi *class multi-label.* Selama fase training, YOLOV3 menggunakan *binary cross-entropy loss* alih alih menggunakan *general mean square* untuk memprediksi kesalahan *class.*
2) *Different bounding-box prediction*. Skor objektivitas diatur ke 1 jika terikan pada *box* sebelumnya tumpang tindih dengan objek kebenaran lebih besar dari lain. Jika bounding box sebeumnya tumpang tindih dengan objek kebenaran dasar lebih banyak dari ambang yang dipilih, rediksi akan diabaikan. Karena itu, YOLOV3 hanya memiliki satu bounding box anchor untuk setiap kebenaran dasar objek.

3) *Predictions across scale*. YOLOV3 dapat memprediksi *box* di tiga skala yang berbeda dan kemudian mengekstrak fitus dari skala tersebut menggunakan fitur piramida *networks*. Hasil prediksi jaringan ialah tensor 3-d, yang mengkodekan *bounding box, objectness score,* dan *class predictions.*

4) Darknet-53. Darknet-53 menggabungkan ide ide CNN lainnya. Ini ialah CNN berlapis 53 termasuk berturut turut 3x3 dan 1x1 lapisan konvolusional, yang menggunakan koneksi yang melewati *network* yang terinspirasi dari ResNet.

   Hasil eksperimen menunjukkan bahwa YOLOV3 berkinerja baik, dan memiliki lebih sedikit operasi *floating point* dan lebih cepat dalam kecepatan.



Gambar 2.7 YOLO [17]

2.5 Faster-YOLO object detection model

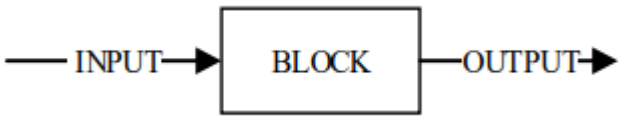
Faster-YOLO digunakan untuk memsimplekan struktur network dan mengurangi komputasi dan kebutuhan *memory*, Faster-YOLO dideskripsikan memakai 4 bagian yaitu: *input-image, feature extraction network, bounding box prediction* dan *final detection result.*

1. *Input Image*. Ukuran *input image* ialah 416 x 416. Dan dipisah menjadi *s x s grid.* Jika bagian tengah object jatuh kedalam *grid cell*, *grid cell* tersebut bertanggung jawab untuk mendeteksi object tersebut
2. *Feature extraction network*. Menggunakan network gabungan dari DRKCELM dan DLELM-AE sebagai fitur mengekstrak dan mengklasifikasi dan mendeteksi.
3. *Bounding box prediction.* Setiap *grid cell* memprediksi B *bounding boxes* dan *confidence scores* untuk setiap *boxes*, setiap *bounding box* bertanggung jawab untuk memprediksi setiap nilai dan *confidence*.
4. *Final detection result. Algoritma The non-maximum suppression* (NMS) digunakan untuk mendeteksi final objek.

Pada hasil eksperimen paper [17] disebutkan bahwa faster yolo menunjukkan hasil yang sama dengan YOLOv3 namun dengan fps yang lebih tinggi.

2.5 Block Diagram

Proses yang terjadi di dalam suatu sistem, dapat digambarkan dengan mudah melalui data masukan, proses, serta keluaran yang dihasilkan. Model yang memiliki ketiga tahapan dinamakan dengan model blok diagram. Blok diagram digunakan untuk merepresentasikan suatu sistem atau sejumlah blok yang berhingga dalam rangkaian beberapa proses menggunakan blok. Elemen yang terdapat pada diagram blok terdiri dari sebab akibat input dan output. Berikut gambaran blok diagram secara umum



Gambar 2.8 Block Diagram

Penggunaan blok diagram pada penelitian ini untuk menggambarkan beberapa proses yang terjadi pada sistem dari mulai data masukan sampai hasil yang diharapkan.

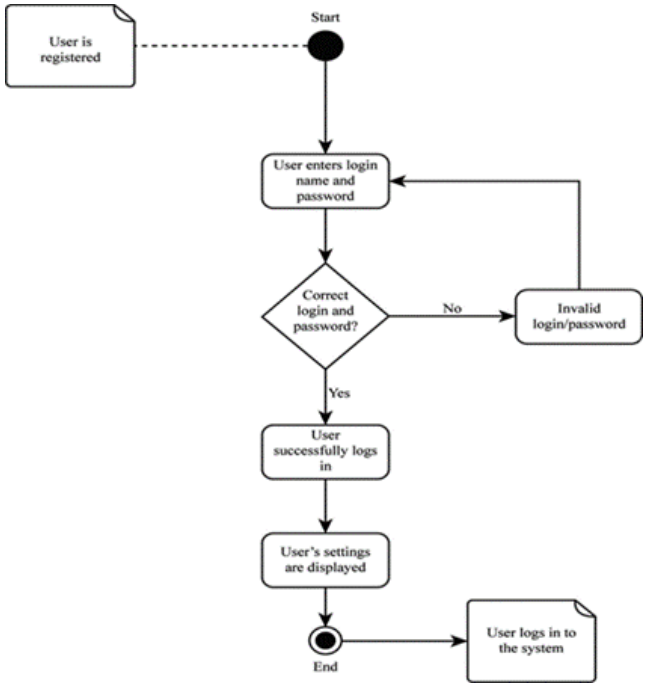
2.6 *Unified Modeling Language*

Unified Modeling Language (UML) adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membanngun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

Unified Modeling Language (UML) adalah sebuah bahasa yang berdasarkan grafik atau gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OO (Object-Oriented). UML sendiri juga memberikan standar penulisan sebuah sistem blue print, yang meliputi konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema database, dan komponenkomponen yang diperlukan dalam sistem software Diagram Unified Modelling Language (UML) antara lain sebagai berikut [18] :

2.6.1 *Activity Diagram*

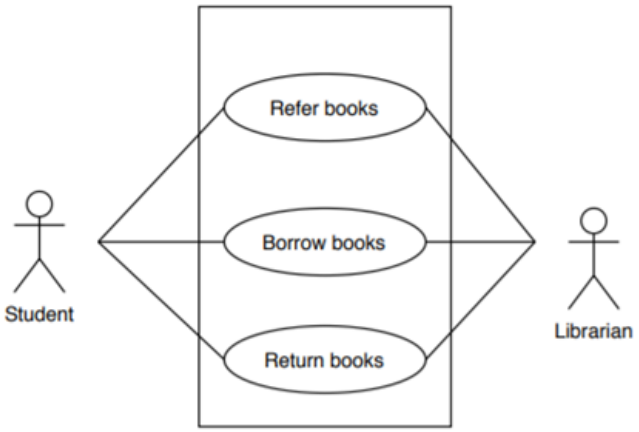
Diagram *activity* menunjukkan aktivitas sistem dalam bentuk kumpulan aksi-aksi, bagaimana masing-masing aksi tersebut dimulai, keputusan yang mungkin terjadi hingga berakhirnya aksi. *Activity* diagram juga dapat menggambarkan proses lebih dari satu aksi salam waktu bersamaan. "*Diagram activity* adalah aktifitas-aktifitas, objek, *state*, transisi state dan *event*. Dengan kata lain kegiatan diagram alur kerja menggambarkan perilaku sistem untuk aktivitas".



Gambar 2.9 Contoh Activity Diagram

2.6.2 *Use Case Diagram*

Use case menggambarkan *external view* dari sistem yang akan kita buat modelnya. *Model use case* dapat dijabarkan dalam diagram *use case*, tetapi perlu diingat, diagram tidak indetik dengan model karena model lebih luas dari diagram. *Use case* harus mampu mengambarkan urutan aktor yang menghasilkan nilai terukur.

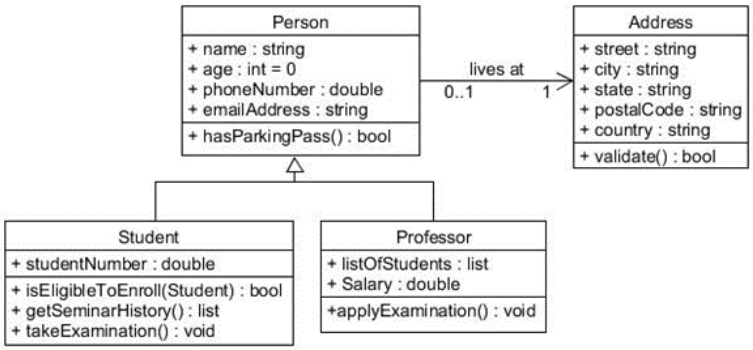


Gambar 2.10 Contoh Use Case Diagram

2.6.3 *Class Diagram*

Kelas sebagai suatu set objek yang memiliki atribut dan perilaku yang sama, kelas kadang disebut kelas objek. *Class* memiliki tiga area pokok yaitu:

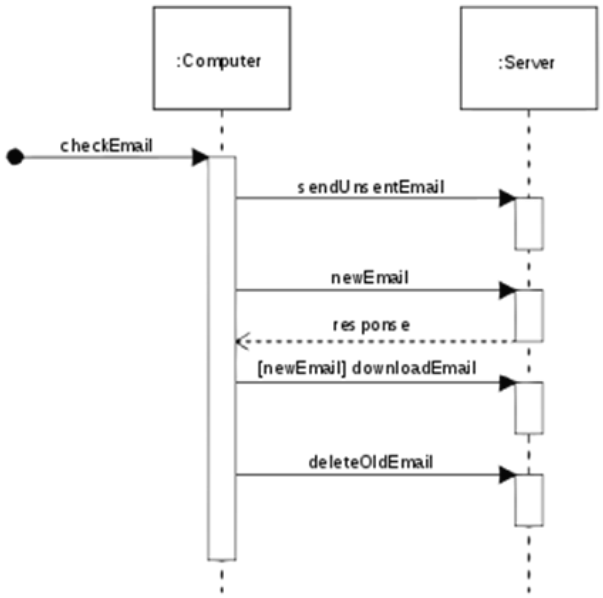
1. Nama, kelas harus mempunyai sebuah nama.
2. Atribut, adalah kelengkapan yang melekat pada kelas. Nilai dari suatu kelas hanya bisa diproses sebatas atribut yang dimiliki.
3. Operasi, adalah proses yang dapat dilakukan oleh sebuah kelas, baik pada kelas itu sendiri ataupun kepada kelas lainnya.



Gambar 2.11 Contoh Class Diagram

2.6.4 Sequence Diagram

Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan *use case* diagram".



Gambar 2.12 Contoh Sequence Diagram 1