

Lampiran 1 : Komponen-Komponen Pemrograman yang Dibutuhkan dan Source-nya di Internet Bila Ada

- Microsoft Visual C++ 6.0.
Program ini terdapat dalam CD paket aplikasi Microsoft Visual Studio 6.0.
- Intel® Image Processing Library (IPL) 2.5.
Filename : ipl25.exe
URL : <http://developer.intel.com/software/products/perflib/ipl/index.htm>
Default folder : C:\Program Files\Intel\plsuite
- Intel® Open Source Computer Vision Library (OpenCV) b3.1.
Ada beberapa versi update dari Intel OpenCV Library, namun yang digunakan dalam proyek tugas akhir ini adalah versi beta 3.1, versi yang telah banyak digunakan karena telah teruji cukup baik digunakan, sedangkan yang terbaru adalah versi beta 4a.
Filename : OpenCV_b3.1.exe
URL : <http://sourceforge.net/projects/opencvlibrary/>
Default folder : C:\Program Files\Intel\OpenCV
Pada salah satu proses instalasinya ada option “add <>\OpenCV\bin to the system PATH”. Pilih option itu untuk mengenali file-file .dll yang dibutuhkan.
- Microsoft DirectX Software Development Kit (DirectXSDK) 8.1.
Filename : DX81SDK_FULL.exe
URL:
<http://download.microsoft.com/download/whistler/dx/8.1/W982KMeXP/>
Default folder : C:\mssdk
(bukan C:\dxsdk karena beberapa sample program akan memanggil pada direktori C:\mssdk).
Nantinya untuk melihat contoh proses kalibrasi kamera perlu menjalankan program GraphEdit setelah instalasi DirectXSDK 8.1, yaitu pada *Start* menu (desktop) → *All programs* → *Microsoft DirectXSDK8.1* → *DirectX Utilities* → *GraphEdit*. Adapun penggunaan GraphEdit adalah sebagai berikut:
 - a. Menu *graph* → *insert filters*.

Lampiran 1 : Komponen-Komponen Pemrograman yang Dibutuhkan dan Source-nya di Internet Bila Ada (sambungan)

- b. Aktifkan/ tancapkan *webcam*.
 - c. Akan muncul filter baru pada *Video Capture Sources* → *Logitech QuickCam Express*. Insert filter tersebut.
 - d. Expand *DirectShow Filters* dengan mengklik tanda +.
 - e. Insert filter *CalibFilter* serta *Video Renderer (input)*.
 - f. Bila ada filter yang tidak bisa di-insert dengan pesan error '*file module cannot be found*', maka cari file *.ax* (expand tanda + pada filter tersebut untuk melihat nama filenya) di *windows explorer* dan register file *.ax* tersebut dengan menjalankan *regsvr32.exe* di *C:\Windows\System 32*
- Logitech WebCam Driver v730
Filename : Logitech Image Studio v730 enu.exe
Default folder : (otomatis ditentukan oleh program installer)

Lampiran 2 : Setting Komponen yang Diperlukan

1. Copykan directory `\Intel-OpenCV2.1\aplikasi-tambahan\CvIGrFrmts` (dari CDRom) menuju directory `C:\Program Files\Intel\opencv\otherlibs\`
2. Tambahkan *path* instalasi program, yaitu:
 - `c:\program files\intel\opencv\bin; c:\program files\intel\plsuite\bin`
(perhatikan pemakaian karakter ‘;’ untuk memisahkan *path*).
 - Untuk Windows 2000/XP:
Klik *Start* → *Settings* → *Control Panel* → *System* → *Advanced* → *Environment Variables* → *System Variables* → tambahkan pada *Variable Path*.
 - Untuk Windows 98:
Tambahkan *path* pada *file* `autoexec.bat`, yang secara *default* terletak di *root directory*, yaitu di `c:\`
3. Untuk mencoba sampel-sampel OpenCV, execute dulu “`C:\Program Files\Intel\OpenCV\bin\RegisterAll.bat`” untuk meregister semua file `.ax`. Hal ini bisa juga dilakukan satu persatu dengan menjalankan ‘`regsvr32.exe<spasi>file.ax`’ melalui menu *Start* → *Run*. File `regsvr32.exe` terdapat pada folder `C:\Windows\System32\`.
4. **Peringatan:** disarankan untuk tidak menjalankan program `C:\Program Files\Intel\OpenCV\utils\maintainer_clean.cmd` karena akan menghilangkan semua file `.dll` yang dibutuhkan yang ada pada `C:\Program Files\Intel\OpenCV\bin`.

Sedangkan pada Microsoft Visual C+ perlu dilakukan proses berikut:

1. Tambahkan file-file `.lib` berikut dengan klik menu *Project* → *Settings* → *Link* → *Object/library Modules*:
 - `vwf32.lib cv.lib highgui.lib ipl.lib cvcam.lib`
(tiap file `.lib` dipisahkan oleh spasi).
2. Dahului dengan melakukan *build baseclasses* dari *DirectShow* baik untuk versi *Win32 Release* maupun *Debug*, yaitu dengan membuka *file* `baseclasses.dsp` pada lokasi:

Lampiran 2 : Setting Komponen yang Diperlukan (sambungan)

C:\MSSDK\samples\Multimedia\DirectShow\BaseClasses\
melalui Visual C++ dengan memilih menu *Build* → *Batch Build*.

3. Pada menu *Tools* → *Options* → *Directories* tambahkan file-file berikut pada path *include*, *library*, dan *executable*. Bila ada file *include*, *library*, atau *executable* yang lain, semisal *include* milik Microsoft Visual Studio, letakkan file milik *opencv* dan *ipl* ke posisi paling atas untuk menghindari konflik atas pemakaian nama file yang sama.

Include files:

C:\PROGRAM FILES\INTEL\OPENCV\CV\INCLUDE
C:\PROGRAM FILES\INTEL\OPENCV\CVAUX\INCLUDE
C:\PROGRAM FILES\INTEL\OPENCV\APPS\COMMON
C:\PROGRAM FILES\INTEL\OPENCV\OTHERLIBS\HIGHGUI
C:\PROGRAM FILES\INTEL\PLSUITE\INCLUDE
C:\MSSDK\SAMPLES\MULTIMEDIA\DIRECTSHOW\BASECLASSES
C:\MSSDK\SAMPLES\MULTIMEDIA\COMMON\INCLUDE
C:\MSSDK\INCLUDE

Library files:

C:\PROGRAM FILES\INTEL\OPENCV\LIB
C:\PROGRAM FILES\INTEL\OPENCV\OTHERLIBS\HIGHGUI
C:\PROGRAM FILES\INTEL\OPENCV\OTHERLIBS\CVLGRFMTS
C:\PROGRAM FILES\INTEL\PLSUITE\LIB\MSVC
C:\MSSDK\SAMPLES\MULTIMEDIA\DIRECTSHOW\BASECLASSES\RELEASE
C:\MSSDK\SAMPLES\MULTIMEDIA\DIRECTSHOW\BASECLASSES\DEBUG
C:\MSSDK\LIB

Executable files:

C:\PROGRAM FILES\INTEL\OPENCV\BIN
C:\PROGRAM FILES\INTEL\PLSUITE\BIN

Lampiran 2 : Setting Komponen yang Diperlukan (sambungan)

4. Tambahkan direktori file `camera.h` dan `matrix.h` pada *Tools* → *Options* → *Include Files*, atau tambahkan perintah berikut ke dalam project source file (*MainFrm.cpp*):

```
#include <camera.h>
```

```
#include <matrix.h>
```

MANUAL PEMAKAIAN SOFTWARE

1. Pada folder "GAMBAR" copykan file-file berikut ke folder "PROGRAM/Copy of 021 – v26 (result of Jesus-final)/appku":
 - c0000.jpg
 - c0001.jpg
 - c0002.jpg
 - c0003.jpg
 - c0004.jpg
 - c0005.jpg
 - c0006.jpg
 - c0007.jpg
 - c0008.jpg
 - c0009.jpg
 - c0001-objek.jpg
 - c0001.jpg
 - contoh-bola-merah.jpg
 - bola1.jpg
 - photo11.jpg
2. Pada folder "PROGRAM/Copy of 021 – v26 (result of Jesus-final)/appku" jalankan file workspace "appku.dsw".
3. Lakukan setting seperti pada folder "BUKU TA/Appendices.pdf".
4. Running program.
5. Bila fungsi kamera tidak bisa berjalan, lakukan 1 saja dari 2 hal berikut:
 - a. copykan file-file berikut dari "PROGRAM/Copy of 021 – v26 (result of Jesus-final)/CCamera" ke folder "PROGRAM/Copy of 021 – v26 (result of Jesus-final)/appku":
 - camera.cpp
 - camera.h
 - b. pada microsoft visual c++ pilih menu "Project? Add to project? Files", lalu pilih ke 2 file camera di atas.
6. Penjelasan fungsi tombol-tombol ada pada Chapter 5.pdf pada folder BUKU TA.
7. Berikut adalah urutan menjalankan program:

- a. Klik tombol 'manual image' untuk memasukkan ke 10 gambar pertama pada no 1.
- b. Bila diinginkan untuk merubah nilai input klik 'input'.
- c. Klik 'calibrate' dan selanjutnya tekan enter pada gambar yang tampil untuk melihat keseluruhan proses.
- d. Klik 'initial' dan drag mouse untuk menggambar persegi pada window camshift pada gambar lingkaran merah.
- e. Klik 'track' dan tekan enter. Selanjutnya tekan enter pada window camshift dan ikuti intruksi yang tampil. Biarkan parameter pada window camshift bernilai default.
- f. Klik 'matrix'.
- g. Klik 'free 2D' dan klik pada gambar untuk melihat hasil titik 2D-nya.
- h. Pada main window output (sebelah kanan), klik tombol sesuai intruksi pada namanya (misal: 'after matrix' berarti tombol baru bisa berfungsi setelah mengklik tombol 'matrix').
- i. Pada urutan ke a., mendapatkan gambar bisa dengan ke 4 cara yang lain.
 1. klik 'capture' untuk mendapatkan 1 gambar dari saat frame berjalan.
 2. klik 'auto capture' untuk mendapatkan sejumlah gambar berurutan, dan klik lagi bila selesai.
 3. klik 'dot' untuk menggambar sendiri titik-titik kalibrasi.
 4. klik 'lens' setelah klik 'calibrate', kemudian klik 'calibrate' lagi untuk mengkalibrasi gambar yang berbeda hasil undistorsi.

1. Judul

Aplikasi Bird View untuk Pemetaan Objek pada Ruangan.

2. Latar Belakang dan Permasalahan

Beberapa peristiwa yang terjadi seringkali memerlukan pengamatan jarak tertentu untuk kepentingan pemetaan objek karena beberapa hal, semisal karena keterbatasan tempat untuk melakukan pengamatan hingga perlu dilakukan dari jarak yang lebih jauh dengan memakai teknik pemetaan, atau pengamatan yang memang lebih baik bila dilakukan dari jarak jauh. Contoh penerapannya antara lain yaitu untuk mengawasi koleksi museum yang tersimpan di dalam ruangan yang tertutup rapat, atau pengamatan terhadap aktivitas pergerakan hewan di dalam ruangan yang bersekat-sekat atau yang disebut labirin.

Selain itu informasi hasil pengamatan diperlukan untuk dicatat dan ditampilkan kembali untuk pengamatan dan pengembangan lebih lanjut atau untuk tujuan pelaporan. Informasi itu antara lain posisi objek, perpindahannya, luas ruangan dan lain-lain.

3. Perumusan Masalah

Untuk pelaksanaan aktivitas itu dibutuhkan perangkat pengamat untuk mengambil gambar objek yang dilengkapi dengan aplikasi untuk mengolah informasi hasil pengamatan serta memindahkannya ke dalam bentuk 2 dimensi (2D) pada layar komputer. Dengan berpatokan pada tujuan itu digunakan sebuah webcam yang diletakkan pada posisi statis dengan jarak tertentu dan posisi sembarang (tidak harus tegak lurus sumbu koordinat tertentu) terhadap objek.

Aplikasi yang dibuat ini nantinya akan menampilkan objek yang diamati dalam representasi 2D, dimana pengamatan terutama hanya ditujukan pada posisi dan gerak perpindahan objek dalam ruangan yang bersangkutan. Dengan memanfaatkan koordinat x dan y dari objek terhadap ruangan, aplikasi akan memperoleh informasi mengenai keberadaan objek dan memetakannya di layar komputer disertai informasi lainnya yang dibutuhkan.

4. Ruang Lingkup

Adapun batasan-batasan dalam pelaksanaan tugas akhir ini meliputi :

- a. Webcam hanya dapat menangkap gambar objek yang berada dalam daerah visualnya meskipun objek masih berada pada 1 ruangan yang sama.
- b. Ruangan tidak bersekat-sekat penuh, yang berarti objek dan webcam masih berada pada 1 bidang ruangan yang sama.
- c. Membutuhkan pencahayaan yang cukup pada objek dan ruangan yang diamati.
- d. Objek bergerak hanya pada batas kecepatan tertentu.
- e. Jumlah webcam 1 buah.
- f. Jumlah objek pengamatan untuk pengujian 1 buah.
- g. Memerlukan proses kalibrasi untuk webcam terlebih dahulu.

Teknologi yang dipakai untuk menciptakan kecerdasan buatan adalah :

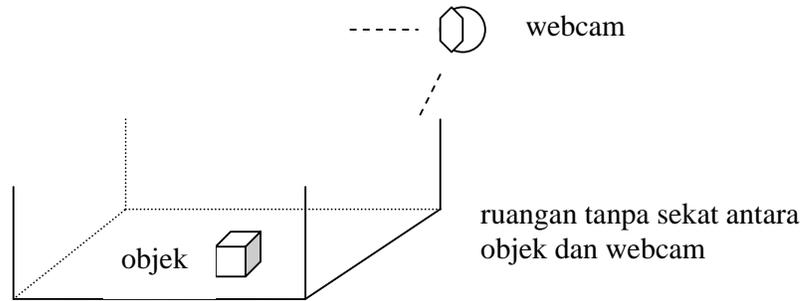
- a. Platform pemrograman : Microsoft Visual C++.
- b. Sistem operasi : Microsoft Windows 98.
- c. Aplikasi untuk kepentingan laporan akhir : Microsoft Word 2000.

5. Tujuan Tugas Akhir

Tugas Akhir ini ditujukan untuk menghasilkan suatu aplikasi pengamatan objek di dalam suatu ruangan dengan tujuan memperoleh informasi yang cukup akurat, terlepas dari kekurangannya yang hanya menampilkan pengamatan dalam bentuk representasi 2D. Selain itu agar aktifitas pengamatan dapat dilakukan tanpa kehadiran subjek pengamatnya dan memakai ruang yang jauh lebih kecil untuk pengamat karena alat pengamat berupa webcam.

6. Tinjauan Pustaka

Pada pengambilan gambar oleh webcam ada banyak parameter yang akan mempengaruhi hasil tampilannya pada saat dilihat di layar monitor. Untuk mendapatkan hasil yang akurat maka perlu mengkalibrasikan webcam tersebut. Kalibrasi webcam ini berfungsi untuk menetapkan karakteristik optikal dan geometri internal webcam (parameter intrinsik) serta orientasi dan posisi 3D dari frame webcam yang relatif terhadap sistem koordinat dunia tertentu (parameter ekstrinsik).

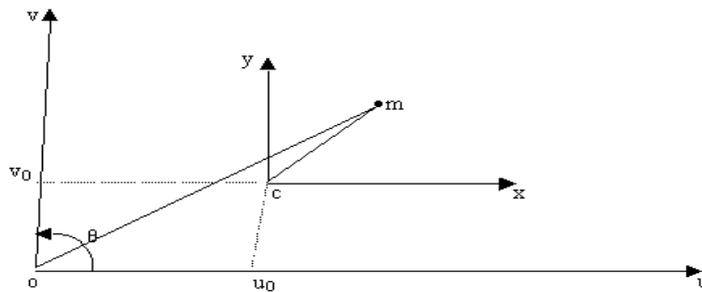


Gambar 6.1. Proses image capture oleh webcam

6.1. Parameter Intrinsik

Parameter intrinsik dari webcam, antara lain :

- Jarak fokus, yaitu jarak antara lensa webcam dan gambar.
- Letak gambar dalam koordinat pixel.
- Efektifitas besar pixel.
- Koefisien distorsi radial dari lensa.



Gambar 6.2. Transformasi koordinat image ke koordinat kamera

Empat parameter diatas menyatakan hubungan antara koordinat pixel pada frame image representasi dan frame webcam. Poros sumbu dari kamera idealnya harus tegak lurus, tapi pada kenyataannya malah membentuk sebuah sudut θ , yang sedikit berbeda dengan 90° . Ini merupakan salah satu parameter intrinsik yang harus diperkirakan (koefisien kemiringan). Selanjutnya kita juga perlu memperkirakan titik asal dari bidang gambar (u_0, v_0) , dimana hal ini tidak selalu bertepatan dengan titik potong poros optik dan bidang gambar. Satuan pada koordinat pixel tidak sama dengan pada titik koordinat m .

Kita misalkan P adalah sebuah titik pada koordinat ruang vektor $XXc = [Xc; Yc; Zc]$ pada frame webcam. Kemudian kita proyeksikan titik tersebut pada

bidang gambar sesuai parameter intrinsik (fc, cc, α_c, kc). Misalkan Xn adalah proyeksi gambar yang telah ternormalisasi

$$Xn = \begin{bmatrix} Xc / Zc \\ Yc / Zc \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

Dengan memisalkan

$$r^2 = x^2 + y^2$$

Setelah ditambahkan dengan distorsi lensa, maka titik koordinat normalisasi yang baru Xd dinyatakan sebagai berikut

$$Xd = \begin{bmatrix} Xd(1) \\ Xd(2) \end{bmatrix} = (1 + kc(1) * r^2 + kc(2) * r^4 + kc(5) * r^6) * Xn + dx$$

dimana $dx = \textit{tangential distortion vector}$

$$dx = \begin{bmatrix} 2 * kc(3) * x * y + kc(4) * (r^2 + 2 * x^2) \\ kc(3) * (r^2 + 2 * y^2) + 2 * kc(4) * x * y \end{bmatrix}$$

Oleh karena itu, ke-5 vektor distorsi (kc) memuat koefisien distorsi baik radial maupun tangensial. (perhatikan bahwa istilah koefisien distorsi radial orde 6 merupakan masukkan kelima vektor kc). Setelah distorsi ditambahkan, koordinat pixel akhir $X_{\text{pixel}} = [Xp; Yp]$ dari proyeksi titik P pada bidang gambar adalah

$$\begin{aligned} Xp &= fc(1) * (Xd(1) + \alpha_c * Xd(2)) + cc(1) \\ Yp &= fc(2) * Xd(2) + cc(2) \end{aligned}$$

Oleh karena itu, koordinat vektor pixel X_{pixel} dan koordinat vektor ternormalisasi Xd saling tergantung satu sama lain melalui persamaan linier berikut.

$$\begin{bmatrix} Xp \\ Yp \\ 1 \end{bmatrix} = KK \begin{bmatrix} Xd(1) \\ Xd(2) \\ 1 \end{bmatrix}$$

dimana KK disini merupakan parameter intrinsik kamera yang didefinisikan sebagai berikut

$$KK = \begin{bmatrix} fc(1) & \alpha_c * fc(1) & cc(1) \\ 0 & fc(2) & cc(2) \\ 0 & 0 & 1 \end{bmatrix}$$

Variabel fc merupakan panjang fokus yang dinyatakan sesuai dalam format matriks dalam matlab (satuanannya mm) yang menunjukkan unit pixel horizontal ($fc(1)=fx$) dan vertical ($fc(2)=fy$) pada gambar. Umumnya besarnya antara fx dan fy adalah sama. Rasio fx/fy disebut "*aspect ratio*", dimana besarnya tidak sama dengan 1 jika pixel dalam CCD array tidak berupa kotak. Oleh karena itu, secara umum model kamera tidak menangani pixel non kotak. Sebagai tambahan bahwa koefisien kemiringan (α_c) menyatakan sudut antara sumbu x dan sumbu y sensor. Konsekuensinya pixel boleh dalam bentuk bukan persegi.

6.2. Parameter Ekstrinsik

Parameter ekstrinsik yang menggambarkan hubungan antara webcam dan dunia nyata, antara lain :

- a. Matriks rotasi.
- b. Vektor translasi.

Dua parameter diatas menyatakan transformasi antara frame webcam dan frame dunia nyata. Hubungan antara titik – titik 3D (M) dan proyeksi gambar pada 2D (m) dinyatakan oleh rumus :

$$m = A[Rt]M$$

$$A = \begin{bmatrix} fx & 0 & cx \\ 0 & fy & cy \\ 0 & 0 & 1 \end{bmatrix} = \text{matrik intrinsik webcam}$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \text{matrik rotasi}$$

$$t = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} = \text{matrik translasi}$$

Keterangan :

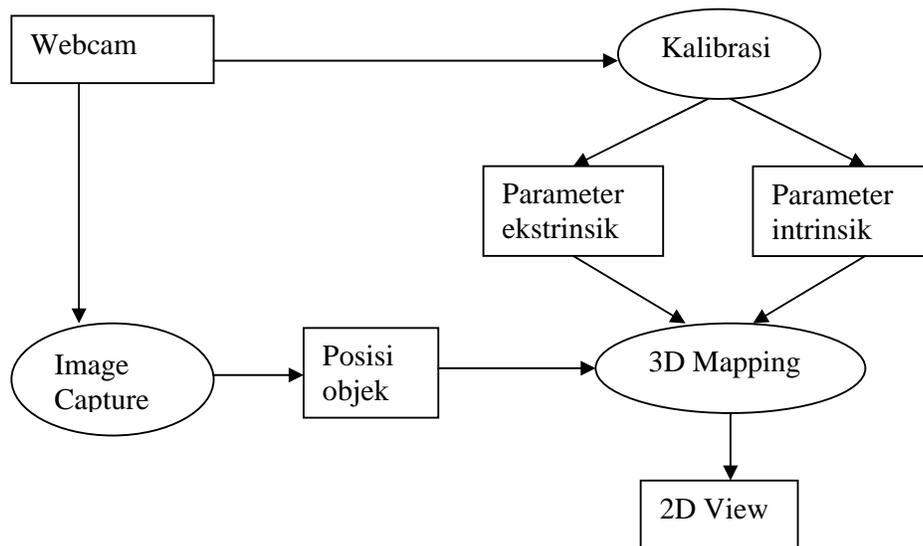
(cx,cy) = koordinat titik pokok

(f_x, f_y) = jarak fokus dalam koordinat x dan y

Berikut ini adalah algoritma yang dipakai untuk kalibrasi webcam :

1. Mencari homography dari semua nilai dalam gambar
2. Menginisialisasi parameter intrinsik; distorsi diberi harga awal = 0
3. Mencari ekstrinsik parameter untuk setiap gambar dalam pola
4. Membuat optimasi dengan meminimalkan error dari nilai proyeksi dari semua parameter

Untuk mengkalibrasikan webcam, maka prosedur kalibrasi ditenun dengan beberapa pola yang telah diketahui ukuran geometrinya dengan pasti, jadi untuk setiap titik dalam pola dan proyeksinya dalam gambar akan melalui prosedur kalibrasi.



Gambar 6.3. Aliran proses dalam pembuatan aplikasi

Setelah webcam melakukan image capture, berikutnya dilakukan tracking posisi untuk menentukan koordinat yang tepat bagi objek yang berada dalam suatu bidang berdasarkan posisi objek tersebut. Hal itu dilakukan dengan cara mencari titik berat dari objek yang di-tracking berdasarkan warna contoh kulitnya. Untuk mencari titik berat dilakukan dengan rumus berikut:

$$W_x = \frac{\sum_{x=0}^k x \times p_x}{jP}$$

$$W_y = \frac{\sum_{y=0}^b y \times p_y}{jp}$$

dimana :

- W_x = koordinat titik berat sumbu x
- W_y = koordinat titik berat sumbu y
- x = koordinat sumbu x yang sedang dihitung
- y = koordinat sumbu y yang sedang dihitung
- p_x = jumlah pixel pada sumbu x
- p_y = jumlah pixel pada sumbu y
- jp = jumlah pixel pada image

Input dari rumus diatas berupa matrik gambar berbentuk *binary* yang hanya berisi nilai 0 (0/hitam) dan 1 (255/putih). Dari rumus itu maka akan diketahui posisi objek pada gambar yang diambil dari kamera. Hasil posisi ini kemudian dikonversi dengan perbandingan resolusi antara layar monitor dan kamera (3D mapping) dan penggabungan ini akan menghasilkan tampilan 2D yang menunjukkan posisi objek dalam sistem koordinat dunia.

7. Metodologi Perancangan

a. Studi Literatur

- Mempelajari teknik kalibrasi untuk webcam.
- Mempelajari teknik pemetaan 3D ke 2D.
- Mempelajari algoritma yang digunakan untuk menentukan posisi obyek dalam hasil representasi 2D.

b. Desain Program

Kegiatan yang dilakukan dalam desain program adalah :

- Desain input dan output. Parameter intrinsik dan ekstrinsik kalibrasi webcam serta gambar objek diperlakukan sebagai inputan aplikasi, sedangkan output berupa representasi 2D objek berikut informasi posisinya.
- Desain algoritma. Menentukan algoritma-algoritma yang dipakai untuk pemetaan 3D ke 2D.

- Desain interface. Interface merupakan sarana penghubung yang memudahkan pemakai memahami dan memakai sistem dalam aplikasi, dimana nantinya akan berupa form yang dioperasikan dengan mouse dan keyboard.

c. Pembuatan Program

Pembuatan aplikasi dengan bahasa visual C++ untuk menerima input dari webcam, mengolah data dengan algoritma, lalu menampilkan ke layar monitor dalam bentuk representasi 2D dan informasi lainnya.

d. Pengujian Program

Pengujian aplikasi dilakukan dengan cara membandingkan antara nilai posisi yang ditampilkan dalam bidang 2D di layar monitor dengan nilai posisi sebenarnya pada bidang 3D.

e. Pengambilan Kesimpulan

Kesimpulan diambil dengan mengetahui sejauh apakah keakuratan informasi yang diperoleh dari proses pengambilan gambar dengan webcam sehingga akan diketahui sejauh apakah aplikasi ini perlu dikembangkan.

8. Jadwal Kegiatan

KEGIATAN	Bulan 1				Bulan 2				Bulan 3				Bulan 4				Bulan 5				Bulan 6			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Studi literature	+	+	+	+	+	+	+	+	+															
Desain program									+	+	+													
Pembuatan program										+	+	+	+	+	+	+	+	+	+	+	+			
Testing program													+				+					+		
Laporan akhir																					+	+	+	+

9. Relevansi

Aplikasi pemetaan ruangan ini dapat dikembangkan lebih lanjut untuk pengamatan jumlah objek yang lebih banyak, seperti pengamatan jalannya pertandingan catur atau bilyar. Untuk menangkap gambar objek yang jauh lebih kecil dapat menggunakan alat pengamat yang lebih baik daripada webcam, namun dengan kendala memiliki harga yang lebih mahal.

10. Daftar Pustaka

- Kochenderfer, Mykel J., Nichol, J. Gordon, Jacobs, Jeffrey Lyman. (2002). The Visual Chess Project. <www.stanford.edu/class/cs223b>
- Kourtis, Lampros, Philiastides, Marios, Stratelos, Constantinos. (2002). A Semi – Empirical Billiards and Pool Tracker. <<http://www.stanford.edu/~kourtis/cs223b/cs223b.htm>>
- Tsai, Roger Y. (1986). An Efficient and Accurate Camera Calibration Technique for 3d Machine Vision. Miami Beach, FL.
- Intel Corporation. (2001). Open Source Computer Vision Library : Reference Manual. U.S.A.: Author. <<http://developer.intel.com>>
- Lenz, R K. Tsai, R Y. (1986). Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3D Machine Vision Metrology. <<http://www.prip.tuwien.ac.at/Research/3DVision/calib.html>>
- Besl, Paul J. McKay, Neil D. (1992). A Method for Registration of 3-D Shapes.