Majalah Ilmiah Mektek

TEKNOLOGI *AUGMENTED REALITY*

Yuri Yudhaswana Joefrie* dan Yusuf Anshori*

Abstract

This journal delivers topic about augmented reality, which is technology that support combination between real world and augmented world. Using webcam to capture the real word and processed using ARToolkitPlus library and the result is generated to the screen.

Keyword: *augmented reality, ARToolkitPlus.*

1. Pendahuluan

Saat ini perkembangan teknologi komputer sudah sedemikian maju. Komputer digunakan di segala bidang kehidupan manusia, mulai dari peralatan rumah tangga sampai bidang pekerjaan rumit.

Teknologi *computer vision* yang merupakan cabang dari kecerdasan buatan atau *artificial intelligence* berkembang sangat cepat. Dengan memanfaatkan komputer, dapat dibuat sangat banyak produk-produk berbasiskan teknologi. Salah satunya yaitu teknologi augmented reality. Augmented reality, yang diterjemahkan bebas berarti realita yang ditambah-tambahkan, merupakan teknologi dari cabang computer vision yang bertujuan untuk menggabungkan citra sintetis ke dalam dunia nyata menggunakan bantuan webcam. Gambar yang ditangkap kemudian diolah dan ditampilkan ke layar monitor. Penggunaan pustaka yaitu *ARtoolkitPlus* memudahkan kita dalam membuat produk berbasiskan augmented reality. Banyak sekali bidang implementasi dari teknologi Augmented Reality ini, contohnya adalah pendidikan, olahraga dan permainan.

2. Tinjauan Pustaka

2.1 Konsep *Augmented Reality*

Aumented Reality (AR) merupakan kebalikan dari Virtual Reality (VR), dimana VR menambahkan obyek nyata didalam dunia maya. Sedangkan konsep AR adalah menambahkan obyek maya ke dalam dunia nyata. Saat perkembangan teknologi semakin meningkat, hal ini juga berpengaruh terhadap bidang *computer vision*. Definisi *computer vision* secara umum adalah merupakan ilmu dan teknologi bagaimana suatu *machine*/sistem melihat sesuatu. Masukan untuk suatu sistem berbasis *computer vision* adalah citra atau *image*. Data citra dapat berbentuk urutan video, citra dari kamera, dan lain-lain.

Beberapa hal yang dikerjakan oleh *computer vision* adalah *recognition*, *motion*, *scene reconstruction*, dan *image restoration*. Berikut beberapa contoh penerapan *computer vision*, yaitu *controlling process*, *detecting events*, *organizing information*, *modeling objects or environtments*, dan *interaction (human-computer interaction)*. AR adalah salah satu teknologi yang menggunakan teknik *computer vision* dalam menetukan kesesuaian antara citra dan dunia nyata, menghitung *pose*, *projection matrix*, homografi dari persesuaian-persesuaian ini.

Kunci kesuksesan dari sistem AR adalah meniru semirip mungkin kehidupan dunia nyata. Dengan kata lain, dari sudut pandang pengguna, pengguna tidak perlu belajar terlalu lama dalam menggunakan sistem AR, sebaliknya, dengan cepat mampu mengoperasikan sistem tersebut berdasarkan pengalaman dalam dunia nyata.

2.2 *ARToolKitPlus*

Merupakan pustaka AR yang secara luas dipakai oleh aplikasi AR di seluruh dunia. Bersifat open source di bawah lisensi GPL dan mudah digunakan. ARToolKitPlus adalah pengembangan

* Staf Pengajar Jurusan Teknik Elaktro Fakultas Teknik Universitas Tadulako, Palu

dari pustaka AR sebelumnya, yaitu ARToolKit yang dikembangkan oleh Hirokazu Kato di Hiroshima City University dan Mark Billinghurst di Human Interface Technology Laboratory (HIT Lab). Saat ini, versi ARToolKit untuk penggunaan non komersil (open source) terhenti di versi 2.72 dan terdapat versi komersil yang terus di-maintain oleh www.artoolworks.com, yaitu versi 4.

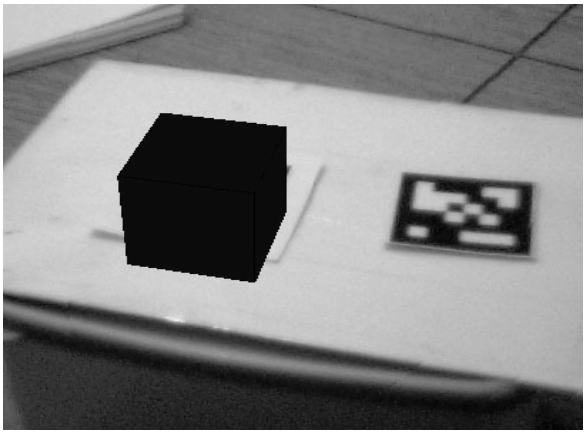
2.3 Pustaka *openGL*

Dalam memunculkan citra sintetis di layar komputer, dibutuhkan pustaka khusus. Dalam hal ini, kita menggunakan pustaka OpenGL yang bersifat cross platform, artinya dapat diimplementasikan pada hampir semua sistem operasi termasuk Windows. Alternatif lain selain OpenGL, ada juga pustaka DirectX yang hanya mampu dijalankan di sistem operasi Windows.

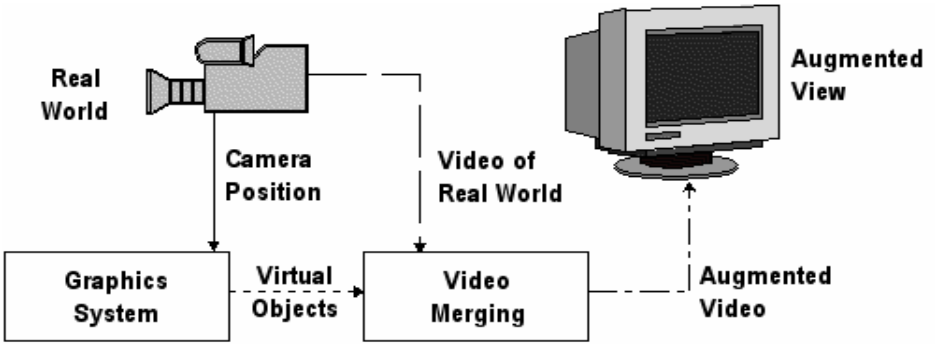
3. Pembahasan

Untuk menjalankan sistem AR, minimal terdiri atas kamera, perangkat monitor, dan dalam kasus-kasus tertentu memerlukan perangkat khusus untuk berinteraksi dengan objek virtual (Gambar 2).

Perangkat monitor dapat diganti dengan perangkat *video see-through* untuk meningkatkan kesan impresif dari objek virtual. Perangkat *video see-through*, biasa juga dinamakan *head-mounted display* (HMD), akan memenuhi seluruh sudut pandang pengguna, sehingga kesan nyata dapat tercapai.

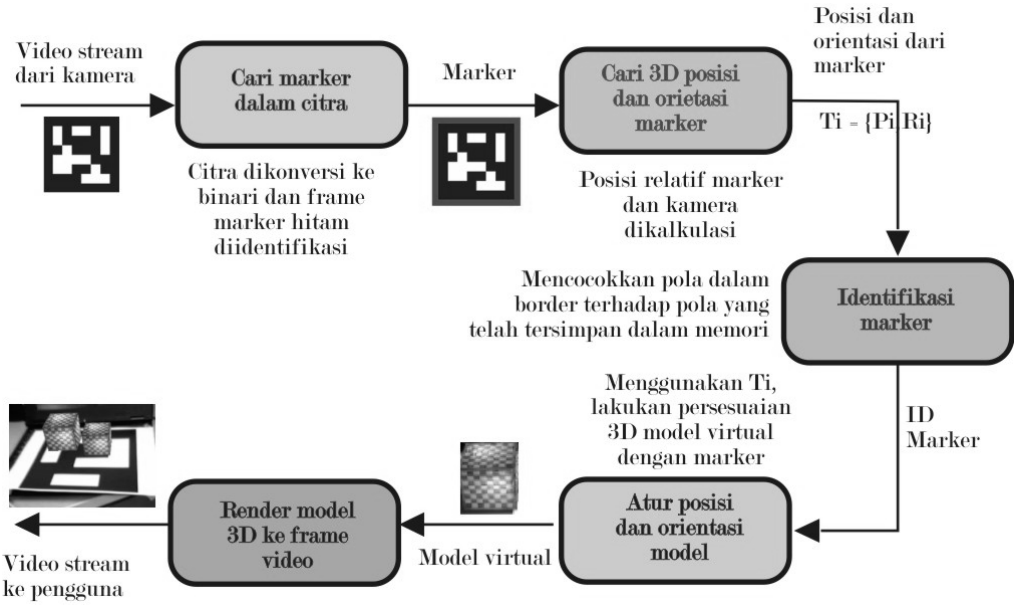


Gambar 1. Model virtual yang berdiri tepat di atas *marker*.



Gambar 2. Perangkat pendukung teknologi AR**Error! Reference source not found.**.

Gambar 3. Penggunaan HMD dalam sistem AR.



Gambar 4. Langkah - langkah untuk me-*render* objek virtual dalam dunia nyata

Untuk menggambar objek virtual dalam dunia nyata, terdapat lima langkah seperti yang ditunjukkan dalam Gambar 3. Pertama, hasil tangkapan citra dari kamera (webcam) diubah dalam bentuk binari (hitam atau putih) berdasarkan nilai *threshold* cahaya. Dalam citra ini kemudian dilakukan pencarian terhadap pola kotak. Kemungkinan ada beberapa kotak yang dikenali dalam tahap ini, namun tidak semua kotak tersebut adalah *marker*. Untuk setiap kotak yang terdeteksi, dilakukan kesesuaian terhadap marker yang sudah dilatih sebelumnya. Jika sesuai, maka ARToolKitPlus menemukan marker tracking atau fiducial marker. ARToolKitPlus kemudian menggunakan ukuran marker dan pola orientasi yang telah diketahui untuk menghitung posisi kamera relatif terhadap marker. Hasil dari perhitungan tersebut dimasukkan ke dalam matriks

3×4. OpenGL kemudian digunakan untuk me-render objek virtual berdasarkan matriks 3×4 yang berisi nilai posisi kamera relatif terhadap marker dalam real world coordinates.

Berikut ini disajikan potongan listing program bahasa C untuk menampilkan citra sintetis hewan dalam dunia nyata menggunakan bahasa pemrograman C++.

```
short initGL(void)
{
        glShadeModel(GL_SMOOTH);
           glClearColor(0.0f, 0.0f, 0.0f, 0.5f);
        glClearDepth(1.0f);
glEnable(GL_DEPTH_TEST);
        glDepthFunc(GL_LEQUAL);
           glHint(GL_PERSPECTIVE_CORRECTION_HINT, GL_NICEST);
        glEnable(GL_TEXTURE_2D);
        return 0;
}

/*
 rutin inisialisasi model
*/
void initModel()
{
        //inisialisasi model 3ds
        printf("Inisialisasi IKAN...\n");
        mdlFish.Init_3ds("data/model/AMBLY_L.3DS");

        printf("Inisialisasi GAJAH...\n");
        mdlElephant.Init_3ds("data/model/ELEPHANT_M.3DS");

        //printf("Inisialisasi PAUS...\n");
        //mdlWheel.Init_3ds("data/model/KWHF_L.3DS");

        //printf("Inisialisasi KURA-KURA...\n");
        //mdlTurtle.Init_3ds("data/model/TURTLE_L.3DS");

        //printf("Inisialisasi ZEBRA...\n");
        //mdlZebra.Init_3ds("data/model/ZEBRA_L.3DS");
        //inisialisasi selesai
}

/*
 rutin inisialisasi tracker
*/
short initTracker()
{
        MyLogger logger;

        //buat turunan tracker
        m_tracker = new ARToolKitPlus::TrackerMultiMarkerImpl<6,6,6,
ARToolKitPlus::PIXEL_FORMAT_RGB>(screen_width, screen_height);
```

```
	//print di console versi artoolkitplus
	printf("ARToolKitPlus v%d.%d \n\n", ARTOOLKITPLUS_VERSION_MAJOR,
ARTOOLKITPLUS_VERSION_MINOR);

	//set logger
	m_tracker->setLogger(&logger);

	//set pixel format RGB
	m_tracker->setPixelFormat(ARToolKitPlus::PIXEL_FORMAT_RGB);

	//inisialisasi tracker
	//parameter:
	//1-> kalibarsi kamera
	//2-> berkas konfigurasi marker yang digunakan
	//3-> near clip atau lingkup area terdekat
	//4-> far clip atau lingkup area terjauh
	if(!m_tracker->init("data/marker/LogitechPro4000.dat", "data/marker/marker_animalpedia.cfg",
1.0f, 1000.0f))
	{
		return -1;
	}

	//set ketebalan border
	m_tracker->setBorderWidth(0.125f);

	//mengaktifkan autothreshold
	m_tracker->activateAutoThreshold(true);

	//mengaktifkan deteksi normal
	m_tracker->setUseDetectLite(false);

	// let's use lookup-table undistortion for high-speed
	// note: LUT only works with images up to 1024x1024
	m_tracker->setUndistortionMode(ARToolKitPlus::UNDIST_LUT);

	//mode marker yang digunakan adalah simple marker
	m_tracker->setMarkerMode(ARToolKitPlus::MARKER_ID_SIMPLE);

	//aktifkan mode pose estimator: RPP
	m_tracker->setPoseEstimator(ARToolKitPlus::POSE_ESTIMATOR_RPP);

	return 0;
}

/*
 rutin inisialisasi kamera/webcam
*/

int initCam()
{
	m_pWebCam = new CWebCam();
	m_pWebCam->initCamera(0,10, screen_width, screen_height);
	return 0;
```

```
}

/*
 rutin untuk meng-handle saat pengguna mengubah ukuran layar opengl
*/

void resize (int width, int height)
{
	screen_width=width;
	   screen_height=height;

	glClear (GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
	glViewport(0,0,screen_width,screen_height);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluPerspective(45.0f,(GLfloat)screen_width/(GLfloat)screen_height,1.0f,10000.0f);

	glutPostRedisplay (); // This command redraw the scene (it calls the same routine of
glutDisplayFunc)
}

/*
 rutin render model 3d di tengah
*/
void renderDiTengah(int idx)
{
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();

	//blok argDraw3dCamera
	glViewport(0, 0, screen_width, screen_height);
	glMatrixMode(GL_PROJECTION);
	glLoadMatrixf( m_tracker->getProjectionMatrix() );

	glEnable(GL_DEPTH_TEST);
	glDepthFunc(GL_LEQUAL);

	/* load the camera transformation matrix */
	glMatrixMode(GL_MODELVIEW);
	glLoadMatrixf( m_tracker->getModelViewMatrix() );

	glMatrixMode(GL_MODELVIEW);

	//render 3ds - weks,,,
	switch (idx)
	{
		case 0:
			glScalef(0.2f, 0.2f, 0.2f);
			mdlFish.Render_3ds();
			break;
		case 1:
			glScalef(0.07f, 0.07f, 0.07f);
			mdlElephant.Render_3ds();
```

```
                        break;
                case 2:
                        glScalef(0.07f, 0.07f, 0.07f);
                        glRotatef(-90.0f, 0.0f, 1.0f, 0.0f);
                        glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
                        mdlWheel.Render_3ds();
                        break;
                case 3:
                        glScalef(0.07f, 0.07f, 0.07f);
                        mdlTurtle.Render_3ds();
                        break;
                case 4:
                        glScalef(0.04f, 0.04f, 0.04f);
                        mdlZebra.Render_3ds();
                        break;
        }

        glDisable( GL_DEPTH_TEST );
}

void argConvGlpara( ARFloat para[3][4], ARFloat gl_para[16] )
{
   int     i, j;

   for( j = 0; j < 3; j++ ) {
      for( i = 0; i < 4; i++ ) {
         gl_para[i*4+j] = para[j][i];
      }
   }
   gl_para[0*4+3] = gl_para[1*4+3] = gl_para[2*4+3] = 0.0;
   gl_para[3*4+3] = 1.0;
}


/*
 rutin render model 3d di tengah spesifik marker
*/
void renderDiTengahMarker(int idx, ARFloat trans[3][4])
{
        ARFloat    gl_para[16];

        glMatrixMode(GL_MODELVIEW);
        glLoadIdentity();

        //blok argDraw3dCamera
        glViewport(0, 0, screen_width, screen_height);
        glMatrixMode(GL_PROJECTION);
        glLoadMatrixf( m_tracker->getProjectionMatrix() );

        glEnable(GL_DEPTH_TEST);
        glDepthFunc(GL_LEQUAL);

  /* load the camera transformation matrix */
```

```
        argConvGlpara(trans, gl_para);
        glMatrixMode(GL_MODELVIEW);
 glLoadMatrixf( gl_para );
        //glLoadMatrixf( tracker->getModelViewMatrix() );

        glMatrixMode(GL_MODELVIEW);

        //render 3ds - weks,,,
        switch (idx)
        {
                case 0:
                        glScalef(0.2f, 0.2f, 0.2f);
                        mdlFish.Render_3ds();
                        break;
                case 1:
                        glScalef(0.07f, 0.07f, 0.07f);
                        mdlElephant.Render_3ds();
                        break;
                case 2:
                        glScalef(0.07f, 0.07f, 0.07f);
                        glRotatef(-90.0f, 0.0f, 1.0f, 0.0f);
                        glRotatef(90.0f, 0.0f, 0.0f, 1.0f);
                        mdlWheel.Render_3ds();
                        break;
                case 3:
                        glScalef(0.07f, 0.07f, 0.07f);
                        mdlTurtle.Render_3ds();
                        break;
                case 4:
                        glScalef(0.04f, 0.04f, 0.04f);
                        mdlZebra.Render_3ds();
                        break;
        }

        glDisable( GL_DEPTH_TEST );
}


/*
 rutin dispaly ke pengguna
*/

void display(void)
{
        ARToolKitPlus::ARMarkerInfo *marker_info = NULL;

        glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
        glLoadIdentity();

        if(dataPtr!=NULL)
        {
                glDrawPixels(screen_width,    screen_height,    GL_BGR_EXT,GL_UNSIGNED_BYTE,
dataPtr_);
```

```
		//here we go, just one call to find the camera pose
		int numDetected = m_tracker->calc(dataPtr);

		if (numDetected > 0)
		{
			for ( int i = 0; i<numDetected; i++ )
			{
				//hewan 1
				if ((m_tracker->getDetectedMarker(i)).id == 303)
				{
					marker_info = (ARToolKitPlus::ARMarkerInfo *) &m_tracker-
>getDetectedMarker(i);
					m_tracker->arGetTransMat(marker_info, myobject[0].center,
20, myobject[0].marker_trans);
					renderDiTengahMarker(0, myobject[0].marker_trans);
				}

				//hewan 2
				if ((m_tracker->getDetectedMarker(i)).id == 295)
				{
					marker_info = (ARToolKitPlus::ARMarkerInfo *) &m_tracker-
>getDetectedMarker(i);
					m_tracker->arGetTransMat(marker_info, myobject[1].center,
20, myobject[1].marker_trans);
					renderDiTengahMarker(1, myobject[1].marker_trans);
				}
				//hewan 3
				if ((m_tracker->getDetectedMarker(i)).id == 2)
					renderDiTengah(2);

				//hewan 4
				if ((m_tracker->getDetectedMarker(i)).id == 3)
					renderDiTengah(3);

				//hewan 5
				if ((m_tracker->getDetectedMarker(i)).id == 4)
					renderDiTengah(4);
			}
			//printf("\n\n%d SINGLE marker diketahui.\n  ", numDetected);
		}
	}

	glFlush();					//Finish rendering
	glutSwapBuffers();
}

void timeOut(int id)
{
	m_pWebCam->grabFrame();

	if ((dataPtr_ = (ARUint8 *) m_pWebCam->getFrame()) != NULL)
	{
```

```
                dataPtr = m_pWebCam->getFrameFlipped();
                glutPostRedisplay();
                glutTimerFunc(100,timeOut,0);
        }

}

int main(int argc, char **argv)
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);
        glutInitWindowSize(screen_width,screen_height);
        glutInitWindowPosition(0,0);
        glutCreateWindow("Contoh AR);
           glutTimerFunc(100,timeOut,0);
        glutDisplayFunc(display);
        glutReshapeFunc (resize);

        initCam();
        initTracker();
        initGL();
        initModel();
        glutMainLoop();

        return 0;
}
```

4. Kesimpulan

Pemanfaatan teknologi augmented reality yang maksimal dapat membantu kita dalam berbagai hal. Sebagai contoh, pada pertandingan bulutangkis yang disiarkan di televisi, kita dapat melihat penerapan teknologi augmented reality. Pada saat pertandingan berlangsung, kita dapat melihat citra sintetis ditambahkan di tengah lapangan lapangan. Citra sintetis tersebut biasanya berupa iklan dari sponsor. Pemanfaatan AR dibidang pendidikan juga sangat luas. Misalnya untuk pengenalan bentuk-bentuk hewan pada anak balita.

5. Daftar Pustaka

Yuri Yudhaswana, 2008, “Tangan Virtual Sebagai Model Interaksi Langsung Untuk AR-Residential Area Design”, STEI, ITB Bandung.