

GERAKAN NPC PADA GAME PENGENALAN HURUF MENGGUNAKAN ALGORITMA DIJKSTRA

Nestia Febriyani Putri¹, Anang Kukuh Adisusilo²

^{1,2} Program Studi Teknik Informatika, Fakultas Teknik, Universitas Wijaya Kusuma Surabaya,
nezi.mutz@gmail.com, anang@anang65.web.id

Abstrak

Game sebagai permainan media elektronik yang merupakan sarana hiburan dan pembelajaran bagi segala usia. Tampilan *game* biasanya dibuat menarik dan interaktif. Terutama untuk *game* pembelajaran untuk anak harus menarik dan mudah dipahami dan segi edukasinya. Selain karakter utama yang dimainkan karakter pendamping yang biasa disebut NPC (*non player character*). Dalam mengatur gerakan NPC pada *game* untuk menuju target atau karakter utama biasanya sering mengalami kendala. *Pathfinding* adalah metode yang sangat mendukung untuk mempermudah gerakan NPC dalam mencari target utama. *Pathfinding* yang di dulaing algoritma dijkstra sangat cocok untuk memberi *Artificial Intelegence* pada NPC. Penghitungan algoritma *dijkstra* yang tepat dengan menghitung seluruh area untuk mencari jarak terdekat menuju target. Sesuai *Game* yang berjenis *tower defence* sangat cocok menggunakan algoritma *dijkstra* untuk pergerakan NPC nya di butkitkan dengan lancarnya gerak NPC tanpa mengalami kesalahan dan kelambatan dalam pergerakan NPC tersebut.

Kata Kunci: *game, parnfinding, algorithma dijkstra, tower defence*

Abstract

Game is a game of electronic media is a means of entertainment and learning for all ages. Display dibauat games are usually attractive and interactive maingkin. Especially for learning games for children should be attractive and easy to understand and in terms edukasinya. In addition to the main character played by a companion character commonly called NPC (non-player character). In regulating the movement of NPC in the game to get to the target or the main characters are frequently experienced problems. Pathfinding is the method sanagt support to facilitate the movement of NPC in finding a primary target. Pathfinding algorithm which in dulaing dijkstra very suitable for giving Artificial Intelligence at the NPC. Dijkstra exact calculation algorithm by calculating the entire area to find the shortest distance to the target. In accordance manifold tower defense game is very suitable to use algorithms dijkstra for his NPC movement in butkitkan with smooth motion NPC without experiencing errors and delays in the movement of the NPC.

Keywords: *game, parnfinding, algorithma dijkstra, tower defence*

I. PENDAHULUAN

Game merupakan permainan media elektronik, dan bersifat sebagai sebuah hiburan berbentuk multimedia yang dibuat mearik agar pemain mendapatkan sesuatu agar muncul adanya kepuasan batin. *Game* dapat digunakan sebagai sarana pembelajaran. *Game* dibuat dengan tujuan spesifik sebagai alat pendidikan biasa disebut dengan *game* edukasi, untuk belajar mengenal huruf. Pencipta *game* harus benar-benar memperhitungkan berbagai hal agar *game* dapat mendidik, memberi pengetahuan dan meningkatkan keterampilan. Masyarakat Indonesia bisa menemukan *game* edukasi di berbagai toko buku, pameran atau bazar buku terdekat dan di sekolah-sekolah yang bekerja sama langsung dengan perusahaan pembuat *game* tersebut. Pada *game* dibutuhkan interaksi pemain dengan NPC.

NPC singkatan dari kata "*Non-Playable/Player Character*", yang artinya karakter yang tidak bisa dimainkan. Selain itu supaya *game* dapat menjadi lebih hidup dan menarik dibutuhkan kecerdasan buatan pada *game*.gerakan NPC yang masih kasar dan kurang halus membuat permainan kurang sempurna. Gerakan NPC yang masih berantakan bisa diatasi dengan menggunakan metode *Pathfinding*. *Pathfinding* merupakan masalah yang sering dihadapi oleh industri *game*, oleh karena itu algoritma *pathfinding* telah diteliti selama bertahun-tahun untuk dapat menyelesaikan masalah *pathfinding* lebih maksimal. (Harika Reddy 2013,). Salah satu algoritma yang dapat diterapkan untuk *pathfinding* adalah algoritma dijkstra. Beberapa peneliti telah menggunakan algoritma tersebut untuk menentukan jarak terdekat. Algoritma ini sangat tepat karena penghitungannya yang tepat dalam menentukan rute terpendek suatu NPC untuk mencapai target dengan gerkan yang halus dan

realistik secara visual, namun karena komputasi tidak dilakukan secara real-time, path planning akan sulit diterapkan pada lingkungan yang dinamis. Pada *game* yang berjenis tower defence pergerakan NPC untuk mencapai target sangat cocok untuk melakukan *pathfinding*. *Tower defence* adalah *game* yang dalam permainannya mementingkan pertahanan pada player agar tidak mudah di serang oleh musuh. Pada *game* yang pergerakan NPC tidak menggunakan algoritma *dijkstra* melainkan menggunakan *wandering*, hasil gerakan NPC tidak terarah kecepatan setiap huruf yang muncul berbeda.

II. DIJKSTRA

Dijkstra adalah sebuah algoritma yang diterapkan untuk mencari lintasan terpendek pada grafik atau lintasan berarah. Namun, algoritma ini juga benar untuk *graf* tak berarah. Algoritma *Dijkstra* mencari lintasan terpendek dalam sejumlah langkah. Algoritma ini menggunakan prinsip *greedy*. Sesuai dengan Prinsip *greedy* pada algoritma *dijkstra* bahwa pada setiap langkah kita memilih sisi yang berbobot minimum dan memasukannya dalam himpunan solusi (E.W 1959).

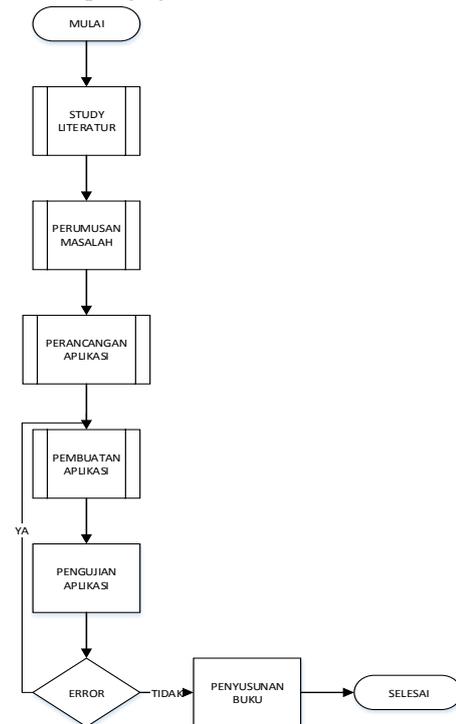
III. METODE

Metodologi yang digunakan dalam penyusunan tugas akhir ini terdiri dari:

1. Studi literatur. Literatur yang digunakan adalah terkait dengan *Artificial Intelligence*, Path finding, Algoritma, unity 3D, bahasa pemrograman C#. Pembelajaran dilakukan dengan cara membaca literatur-literatur yang ada di perpustakaan, mencari di internet, dan bertanya pada orang-orang yang berkompeten di dalamnya.
2. Perumusan masalah. Merumuskan konsep *Artificial intelligence* untuk path finding menggunakan Algoritma Dijkstra serta kebutuhan platform yang dibutuhkan.
3. Perancangan Aplikasi. Merancang model 3d menggunakan blender. Merancang pembuatan scenario permainan. Merancang *storyboard* untuk jalannya *game* beserta NPC nya.
4. Pembuatan Aplikasi *game* pembelajaran huruf menggunakan algoritma *dijkstra*. Pembuatan program dilakukan dengan menggunakan unity3D.
5. Pengujian Aplikasi. Pengujian dilakukan untuk mengetahui tingkat keberhasilan sistem, serta dilakukan evaluasi yang menyangkut konsep dan tema.
6. Penyusunan Buku. Langkah terakhir adalah membuat dokumentasi pelaksanaan tugas akhir yang meliputi teori dasar, proses perancangan, pembuatan, dan hasil pengujian.

3.1. Diagram Alir

Pada sub-bab ini akan disertakan diagram alir yang menggambarkan proses-proses dari riset yang dilakukan oleh penulis. Berikut adalah diagram alir dari pengerjaan.



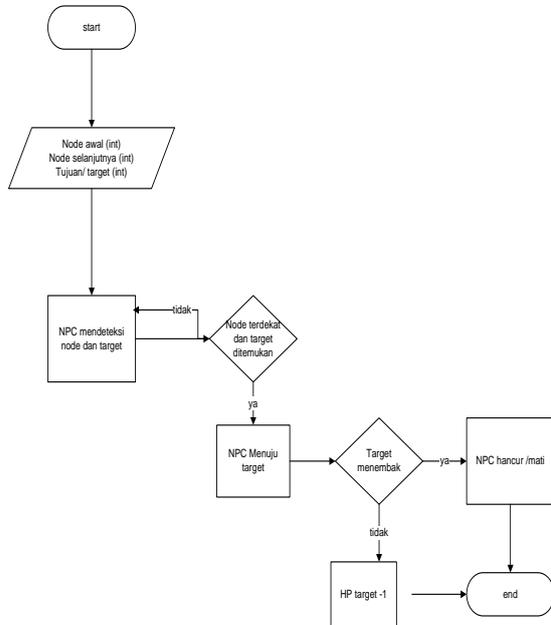
Gambar 1 Diagram Alir

Gambar 1 adalah diagram alir yang menjelaskan proses – proses pembuatan laporan ini mulai dari study literature, perumusan masalah, perancangan aplikasi, pembuatan aplikasi, penyusunan aplikasi, ketika ada eror dalam pengujian aplikasi maka kembali ke proses pembuatan aplikasi. Namun jika tidak terjadi eror lanjut ke tahap berikutnya yaitu penyusunan buku. Semua tahap yang disebutkan memiliki proses lagi yang akan di uraikan. Lambang untuk proses diatas menunjukkan sub proses dari pengerjaan.

IV. PERANCANGAN APLIKASI

4.1. Alur Kerja Aplikasi

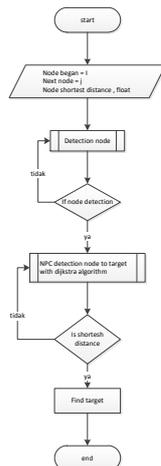
Dalam alur kerja aplikasi ini akan ditunjukkan bagaimana cara kerja NPC untuk menemukan target, dan bagaimana cara NPC untuk menemukan jalur yang terpendek.



Gambar 2. Alur Kerja Aplikasi

4.2. Alur Proses Jalannya NPC menuju target

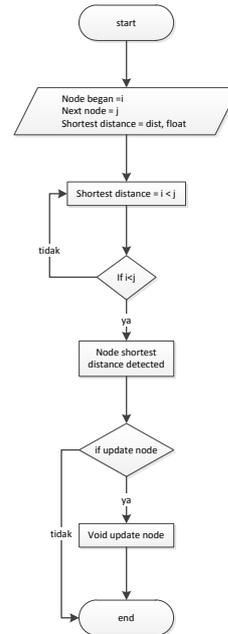
Dalam menuju target NPC akan melalui beberapa tahap untuk menuju target dengan jalur terpendek menggunakan algoritma *dijkstra*.



Gambar 3 Jalannya NPC Menuju Target

4.2.1. Alur Proses Detection Node

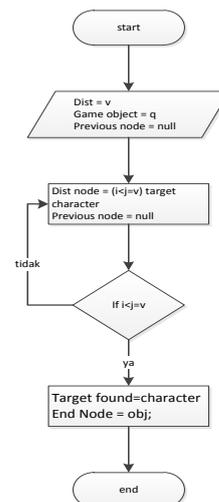
Langkah selanjutnya untuk mendapatkan jalur terpendek untuk menuju target yaitu mendeteksi titik – titik yang telah di buat.



Gambar IV Alur Proses Detection Node

4.2.2. Alur Proses Dijkstra

Langkah selanjutnya untuk mendapatkan jalur terpendek untuk menuju target dengan algoritma *dijkstra* yaitu mendeteksi titik terpendek yang telah di buat.

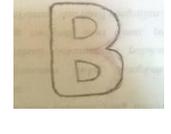
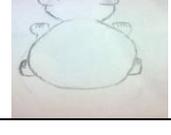


Gambar 5 Alur Proses Dijkstra

4.3. Sketsa Grafic Desain

Dalam perencanaan membuat model 3D pada *game* di butuhkan steksa 2D pada proses perancangan.

Tabel 1 Sketsa Dan Keterangananya

Gambar	Keterangan
	Sketsa dari huruf A yang berperan sebagai NPC Dalam <i>game</i> pengenalan huruf
	Sketsa dari huruf B yang berperan sebagai NPC Dalam <i>game</i> pengenalan huruf
	Sketsa dari huruf C yang berperan sebagai NPC Dalam <i>game</i> pengenalan huruf
	Sketsa dari huruf D yang berperan sebagai NPC Dalam <i>game</i> pengenalan huruf
	Sketsa dari huruf E yang berperan sebagai NPC Dalam <i>game</i> pengenalan huruf
	Karakter panda sebagai player, dan target dari NPC

4.4. Skenario

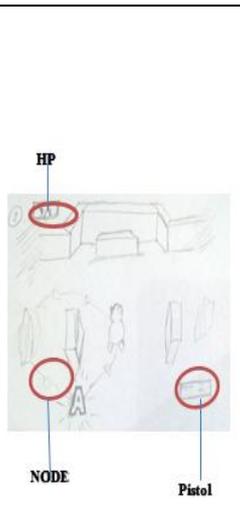
NPC adalah huruf A-E yang berperan untuk menemukan target yang bertujuan untuk merusak pertahanan dari target. Target berlaku sebagai player yang memburu huruf dengan cara menembak huruf secara berurutan dan A, B, C, D, E. Pada awal permainan target berada pada posisinya untuk menunggu NPC datang dan menembaknya sesuai urutan huruf yang ada. 5 detik kemudian NPC datang menuju target untuk berusaha menabrak target. Jika NPC menemui target dan target menembak huruf dengan tepat maka huruf akan hancur dan mati. Namun jika target tidak melakukan

apapun terhadap NPC atau huruf yang datang Health potion pada target akan berkurang dan NPC akan terus mengejar target hingga target tersentuh.

4.5. Storyboard

Storyboard merupakan sketsa atau rancangan gambar yang disusun berurutan sesuai dengan skenario, kita dapat menyampaikan ide cerita kita kepada orang lain dengan lebih mudah dengan storyboard, dengan mengikuti gambar-gambar yang tersaji kita dapat menggiring khayalan seseorang, sehingga menghasilkan persepsi yang sama pada ide cerita kita. Berikut adalah storyboard untuk permainan pengenalan huruf

Tabel 2 Storyboard

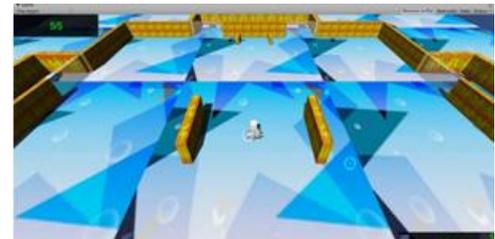
Gambar	Keterangan
	ini adalah tampilan awal pada <i>game</i> , pada detik pertama, kamera berada di atas character, cahanya berada di sudut tampilan, kotak kecil yang terhubung dengan garis adalah rute yg merupakan gabungan node. Titik awal munculnya node. kotak yg berada pada kanan bawah adalah jumlah pitol yang disediakan untuk menembak. Kotak yang berada pada kiri atas adalah jumlah health potion yang dimiliki karakter.
	ini adalah tampilan pada detik ke lima, kamera berada di atas character, cahanya berada di sudut tampilan, kotak kecil yang terhubung dengan garis adalah rute yg merupakan gabungan node. Titik awal munculnya node, huruf A adalah huruf yang pertama muncul dan menghapiri target. kotak yg berada pada kanan bawah adalah jumlah pitol yang disediakan untuk

	menembak. Kotak yang berada pada kiri atas adalah jumlah health potion yang dimiliki karakter.
	<p>pada detik ke 8, kamera mengikuti character, cahayanya berada di sudut tampilan, kotak kecil yang terhubung dengan garis adalah rute yg merupakan gabungan node. Titik awal munculnya node, karakter melihat kemuncululan huruf A dan langung menembak huruf A. kotak yg berada pada kanan bawah adalah jumlah pitol yang disediakan untuk menembak. Kotak yang berada pada kiri atas adalah jumlah health potion yang dimiliki karakter.</p>

	Hasil dari huruf E yang berperan sebagai NPC Dalam <i>game</i> pengenalan huruf
	Karakter panda sebagai player, dan target dari NPC

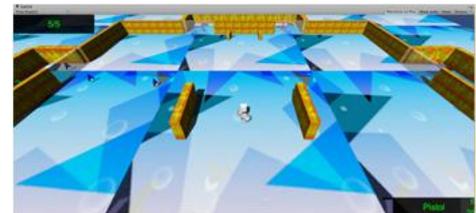
Hasil Aplikasi menggunakan unity 3d game engine

1. Tampilan awal *game*, saat karakter pada posisi diam menunggu huruf / NPC datang menghampiri.



Gambar 6 hasil tampilan awal game

2. Huruf datang menghampiri target mengikuti node yg telah ditentukan.



Gambar 7 gerak NPC dengan dijkstra

VI. HASIL DAN PEMBAHASAN

5.1. Hasil Model 3D menggunakan Blender

Pada tabel 4.5 telahh digambarkan sketsa perancangan objek 3D yang akan di buat. Tabel 5.1 menunjukkan hasil dari sketsa 3D.

Tabel 3 Hasil Objek 3D

Gambar	Keterangan
	Hasil dari huruf A yang berperan sebagai NPC Dalam <i>game</i> pengenalan huruf
	Hasil dari huruf B yang berperan sebagai NPC Dalam <i>game</i> pengenalan huruf
	Hasil dari huruf C yang berperan sebagai NPC Dalam <i>game</i> pengenalan huruf
	Hasil dari huruf D yang berperan sebagai NPC Dalam <i>game</i> pengenalan huruf

5.2. Proses Uji Coba

Pada proses uji coba ini dapat membandingkan hasil gerak NPC yang menggnakan algoritma dijkstra dan yang tidak menggunakan algoritma dijkstra. Pada uji coba kali ini huruf 'A' berwarna biru menunjukkan NPC yang menggunakan algoritma dijkstra. Sedangkan huruf 'a' berwarna hitam menunjukkan NPC yang tidak menggunakan algoritma dijkstra. Dengan perbedaan bentuk dan warna huruf akan lebih mudah membandingkannya. Pada tabel 5.2 dijelaskan perbedaan yang terjadi.

Tabel 4 Tabel Parameter Gerak Huruf

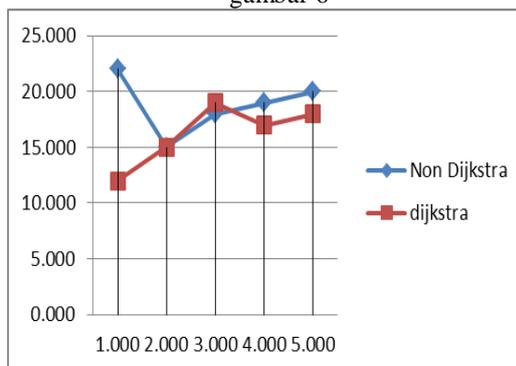
<p>HURUF YANG MENGGUNAKAN ALGORITMA</p>  <p>DIJKSTRA</p>	<ul style="list-style-type: none"> - Kemunculan huruf runtut dan perlahan berjalan mengikuti node. - Kecepatan huruf stabil sesuai yang telah ditentukan. - Jalannya huruf tepat mengikuti node terpendek untuk menuju target. - Ketika target berjalan huruf secara otomatis mengikuti jalannya target.
<p>HURUF YANG MENGGUNAKAN WANDERING</p> 	<ul style="list-style-type: none"> - Kemunculan huruf runtut namun tidak semua huruf yang muncul berjalan mengikuti node. - Kecepatan huruf tidak stabil tidak sesuai yang telah ditentukan. Huruf pertama dan kedua muncul berjalan cepat sekali dan tidak menuju target. - Jalannya huruf tidak tepat mengikuti node untuk menuju target. - Ketika target berjalan huruf secara otomatis mengikuti jalannya target.

Uji coba dilakukan dengan menggunakan FPS(frame per second) sebagai output hasil pengujian. Jumlah bingkai gambar atau frame yang ditunjukkan setiap detik dalam membuat gambar bergerak, diwujudkan dalam satuan fps (frames per second), makin tinggi angka fps-nya disebut Frame rate. Semakin rendah fps nya berarti semakin cepat juga gerakan NPC nya untuk menuju target. Namun sebaliknya jika semakin tinggi FPS nya maka semakin lama dan semakin jauh juga NPC untuk mendekati target.

Tabel 4. Tabel Parameter Gerak Huruf Dalam FPS

Titik (NODE)	Dijkstra	Non Dijkstra	Satuan
1	12.000	22.000	FPS
2	15.000	15.000	FPS
3	19.000	18.000	FPS
4	17.000	19.000	FPS
5	18.000	20.000	FPS

Dari tabel 4 dapat di buat sebuah grafik pada gambar 6



Gambar 6 Grafik Perbandingan

Pada titik terakhir adalah titik dimana huruf menemui target. Pada huruf yang menggunakan algoritma dijkstra yang berada pada titik 5 huruf berada tepat sasaran target. Namun pada huruf yang tidak menggunakan algoritma dijkstra titik 5 menunjukkan bahwa tidak tepat target bahkan berjalan menuju ke arah lain.

VII. PENUTUP

6.1. Kesimpulan

Berdasarkan dari percobaan gerakan NPC pada permainan pengenalan huruf dapat di simpulkan bahwa:

1. Algoritma Dijkstra dapat di implementasikan pada gerakan NPC dalam permainan pengenalan huruf yang berjenis tower defence.
2. Hasil implementasi algoritma dijkstra pada permainan ini menunjukkan keberhasilan NPC dalam menemukan jarak terpendek mengejar target dengan path yang telah di tentukan.
3. Pada NPC yang menggunakan Algoritma dijkstra tidak mengalami kesulitan maupun kesalahan dalam mendeteksi titik terdekat yang telah di tentukan dari keberadaan target.
4. Kekurangannya adalah proses pendeteksian titik oleh NPC yang sedikit lebih lama karena dijkstra menghitung seluruh area terlebih dahulu. Namun dari kekurangan tersebut menghasilkan penghitungan yang akurat.

6.2. Saran

Beberapa saran yang bisa diberikan dalam pengembangan pergerakan NPC pada permainan yang berjenis tower defence ini yaitu :

- Perlu di kembangkan lagi mengenai macam huruf atau NPC pada permainan pengenalan huruf.
- Perlu dikembangkan lagi mengenai halang rintang agar tampilan game lebih menarik.

DAFTAR PUSTAKA

- [1] Admiralty, Great Britain (1915) *Admiralty manual of navigation*, 1914, Chapter XXV: "The Magnetic Compass (continued): the analysis and correction of the deviation", London : HMSO, 525 p.
- [2] Azuma, Ronald. *A Survey of Augmented Reality* Presence: Teleoperators and Virtual Environments, pp. 355–385, August 1997.
- [3] Dijkstra, E.W.: *A Note on Two Problems in Connexion with Graphs*. Numerische Mathematik 1 (1959) 269–271
- [5] Felix Klein and Arnold Sommerfeld, "*Über die Theorie des Kreisels*" (Tr., About the theory of the gyroscope). Leipzig, Berlin, B.G. Teubner, 1898–1914. 4 v. illus. 25 cm.
- [6] Gustavson, Todd (2009). *Camera: a history of photography from daguerreotype to digital*. New York, New York: Sterling Publishing Co., Inc. ISBN 978-1-4027-5656-6.
- [7] Howell, Elizabeth. "Navstar: GPS Satellite Network". SPACE.com. Retrieved February 14, 2013.
- [8] Lester, P., *A* Pathfinding for Beginners*, <http://www.policyalmanac.org/games/aStarTutorial.htm>, tanggal akses 22 Mei 2012.
- [9] Moser, R.: *A Fantasy Adventure Game as a Learning Environment: Why Learning to Program is So Difficult and What Can be Done About It*. SIGCSE Bull. 29(3) (1997) 114–116
- [11] Nakamura, R., Tori, R., Bernardes JR., J., Bianchini,
- [12] R., Jacober, E.: *A Practical Study on the Usage of a Commercial Game Engine for the development of Educational Games*. In: 2nd Games and Digital Entertainment Workshop, Brazilian Computer Society (2003)
- [13] N. N, *Path Finding*, Chapter 12, 1592730051c.Pdf, 2006, tanggal download 22 Mei 2012. *Society of Robots, Introduction to Microcontrollers*, http://www.societyofrobots.com/microcontroller_tutorial.shtml/
- [14] Provatidis, C. G. (2012). Revisiting the Spinning Top, *International Journal of Materials and Mechanical Engineering*, Vol. 1, No. 4, pp. 71–88.
- [15] Sudirman, I., *Perkembangan Software Komputer, Kuliah Pengantar Ilmu Komputer.com*, 2003.
- [16] Richard Hartley and Andrew Zisserman, 2003. *Multiple view geometry in computer vision, 2nd ed. Cambridge University Press*. ISBN 0-521-54051-8.
- [17] Rolland, J; Biocca F; Hamza-Lup F; Yanggang H; Martins R (October 2005). "*Development of Head-Mounted Projection Displays for Distributed, Collaborative, Augmented Reality Applications*". Presence: Teleoperators & Virtual Environments 14 (5): 528–549.

Halaman ini kosong
Redaksi Melek IT