



UNIVERSITAS INDONESIA

**PENGEMBANGAN SISTEM KOMUNIKASI DAN GUI UNTUK
KONTROL ROBOT ARTIKULASI BERBASIS *WEB***

SKRIPSI

**ANOM TEJO PRATOMO
040502012X**

**FAKULTAS TEKNIK
DEPARTEMEN TEKNIK MESIN
DEPOK
DESEMBER 2009**



UNIVERSITAS INDONESIA

**PENGEMBANGAN SISTEM KOMUNIKASI DAN GUI UNTUK
KONTROL ROBOT ARTIKULASI BERBASIS *WEB***

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana teknik

**ANOM TEJO PRATOMO
040502012X**

**FAKULTAS TEKNIK
DEPARTEMEN TEKNIK MESIN
DEPOK
DESEMBER 2009**

Halaman Pernyataan Orisinalitas

Skripsi ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : Anom Tejo Pratomo

NPM : 040502012X

Tanda Tangan :

Tanggal :

Halaman Pengesahan

Skripsi ini diajukan oleh

Nama : Anom Tejo Pratomo
NPM : 040502012X
Program Studi : Teknik Mesin
Judul Skripsi : Pengembangan Sistem Komunikasi dan GUI
untuk Kontrol Robot Artikulasi Berbasis Web

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Mesin, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Dr. Ir. Gandjar Kiswanto, MEng. ()

Penguji : Ir. Hendri D. S. Budiono, MEng. ()

Penguji : Ir. Henky S. Nugroho, MT. ()

Penguji : Dr. Ario Sunar Baskoro, ST, MT., ()
MEng.

Ditetapkan di :

Tanggal :

Abstrak

Nama : Anom Tejo Pratomo
Program Studi : Teknik Mesin
Judul : Pengembangan Sistem Komunikasi dan GUI Untuk Kontrol Robot Artikulasi Berbasis *Web*

Penelitian ini merancang sebuah sistem yang mampu mengontrol sebuah robot artikulasi dengan lima derajat kebebasan dari jarak jauh melalui media *internet* yang berbasiskan aplikasi *web*. Dalam penelitian ini digunakan sebuah komputer yang bertindak sebagai *server* yang dilengkapi dengan dua buah *web camera* untuk memantau kondisi dan pergerakan robot dan juga sebuah mikrokontroler pengontrol robot sebagai pemroses dan pengontrol masukan untuk menggerakkan robot. Melalui sebuah *web browser* pada komputer yang bertindak sebagai *client*, sistem pada komputer *server* diakses oleh pengguna dan menampilkan sebuah *interface* yang dirancang sebagai panel kontrol robot. Melalui *interface* ini pengguna dapat memberi masukan berupa perintah untuk menggerakkan robot yang dapat diberikan dalam dua pilihan mode basis kontrol, yaitu *cursor-based/inverse kinematics* dan *manual/forward kinematics*. Berdasarkan hasil pengujian, sistem mampu menanggapi perintah yang diberikan kemudian memroses dan mengeksekusinya dalam bentuk pergerakan robot sesuai dengan mode dan perintah dari masukan yang diberikan.

Kata kunci:

Kontrol robot, aplikasi web

Abstract

Name : Anom Tejo Pratomo
Study Program : Mechanical Engineering
Title : Development of Communication System and GUI for Web
Based Articulated Robot Control

This research is aimed to design and develop a system capable of remotely controlling a five-degree-of-freedom articulated robot through *internet* platform on a web based application. The research was built with single computer act as a server coupled with a pair of web camera to monitor the status and movement of the robot and also coupled with a robot-controller microcontroller as a processor and controller of inputs to move the robot. Through the web browser on user's computer acting as *client*, the system is accessed by the user and displays an interface designed to be a robot's control panel. Through this interface, user can input command to move the robot which can be given in two different control modes, cursor-based/inverse kinematics and manual/forward kinematics. Based on the result of the experiment, system is able to respond the command then processes and executes it in form of robot movement based on control mode and command of the given input.

Keywords:

Robot control, web application

Ucapan Terima Kasih

Puji syukur penulis panjatkan kepada Allah SWT. yang telah memberikan nikmat serta karunia-Nya, sehingga penulis dapat menyelesaikan laporan skripsi yang berjudul “Pengembangan Sistem Komunikasi dan GUI Untuk Kontrol Robot Artikulasi Berbasis Web”.

Laporan skripsi ini sangat tidak mungkin dapat diselesaikan oleh penulis seorang diri, oleh karena itu ucapan terima kasih ingin penulis sampaikan kepada:

1. Keluarga penulis tercinta yang selalu mendoakan dan menyemangati penulis.
2. Dr. Ir. Gandjar Kiswanto, MEng. selaku pembimbing skripsi dan Sekretaris Departemen Teknik Mesin FTUI.
3. Ir. Hendri D. S. Budiono, MEng. selaku penguji skripsi.
4. Ir. Henky S. Nugroho, MT. selaku penguji skripsi.
5. Dr. Ario S. Baskoro, ST., MT., MEng. selaku penguji skripsi.
6. Teman-teman penulis terutama kedua sahabat penulis, Jelita dan Intan atas bantuan moral dan pemikiran saat penyusunan tulisan ini.
7. Teman-teman mahasiswa teknik mesin UI, khususnya Tim Pengembang i-RoMan: Hendra, Adhyt dan Teguh.
8. Kawan-kawan di komunitas Kaskus UI atas doa dan dukungan morilnya.
9. Kumpulan lagu dari Keane dan Gita Gutawa yang selalu mengiringi penulis dalam mengerjakan skripsi.

Tangerang, 28 Desember 2009

Anom Tejo Pratomo

Daftar Isi

HALAMAN PERNYATAAN ORISINALITAS.....	I
HALAMAN PENGESAHAN	II
ABSTRAK	III
ABSTRACT	IV
UCAPAN TERIMA KASIH.....	V
DAFTAR ISI.....	VI
DAFTAR TABEL.....	VIII
DAFTAR GAMBAR.....	IX
BAB 1. PENDAHULUAN.....	1
1.1 LATAR BELAKANG MASALAH.....	1
1.2 PERUMUSAN MASALAH.....	2
1.3 PEMBatasan MASALAH.....	2
1.4 TUJUAN PENELITIAN	3
1.5 METODOLOGI PENELITIAN.....	3
1.6 SISTEMATIKA PENULISAN.....	4
BAB 2. SISTEM KOMUNIKASI UNTUK KONTROL ROBOT ARTIKULASI BERBASIS WEB.....	5
2.1 PIRANTI LUNAK.....	5
2.1.1 <i>Driver Robot</i>	5
2.1.2 <i>Web Server</i>	6
2.1.3 <i>Web Camera Server</i>	6
2.1.4 <i>Web Browser</i>	6
2.2 PROTOKOL KOMUNIKASI HTTP.....	7
2.3 WEB SCRIPTING	8
2.3.1 <i>Client-side Scripting</i>	8
2.3.2 <i>Server-side Scripting</i>	10
2.4 PIRANTI KERAS.....	12
2.4.1 <i>Server [8]</i>	12
2.4.2 <i>Control Unit</i>	15
2.5 PENGERTIAN MANIPULATOR.....	16
2.5.1 <i>Revolute Joint</i>	18
2.5.2 <i>Prismatic Joint</i>	19
2.5.3 <i>Derajat Kebebasan</i>	19
2.6 BASIS KONTROL ROBOT.....	20
2.6.1 <i>Inverse Kinematics</i>	20
2.6.2 <i>Forward Kinematics</i>	21
BAB 3. PENGEMBANGAN SISTEM KOMUNIKASI DAN GUI.....	22
3.1 LOGIKA SISTEM	23
3.1.1 <i>Inverse Kinematics</i>	23
3.1.2 <i>Forward Kinematics</i>	24
3.2 SPESIFIKASI SISTEM	24
3.2.1 <i>Piranti Keras</i>	24
3.2.2 <i>Piranti Lunak</i>	28
3.3 WEB SCRIPTING	28
3.4 INTERFACE.....	30

3.4.1	<i>Manajemen Sesi dan Proteksi</i>	30
3.4.2	<i>Halaman Kontrol Utama</i>	34
3.4.3	<i>Pengambilan Gambar Web Camera dan Koordinat Cursor Mouse [15]</i>	35
3.4.4	<i>Penanganan Kesalahan</i>	36
3.5	BAGIAN PEMROSESAN	37
3.5.1	<i>Pemrosesan Masukan</i>	37
3.5.2	<i>Struktur Data Keluaran</i>	43
3.6	IMPLEMENTASI SISTEM.....	45
3.6.1	<i>Instalasi</i>	45
3.6.2	<i>Metode Akses</i>	46
BAB 4.	PENGUJIAN DAN ANALISA SISTEM	47
4.1	PERANGKAT UJI COBA	47
4.2	KOMPATIBILITAS WEB BROWSER	48
4.3	STREAMING GAMBAR WEB CAMERA	49
4.4	INVERSE KINEMATICS	49
4.5	FORWARD KINEMATICS	51
4.6	UNJUK KERJA	53
BAB 5.	KESIMPULAN DAN SARAN PENELITIAN LEBIH LANJUT	55
5.1	KESIMPULAN	55
5.2	SARAN PENELITIAN LEBIH LANJUT	55
	DAFTAR REFERENSI	57
	LAMPIRAN	59

Daftar Tabel

Tabel 3-1. Spesifikasi komputer <i>server</i>	24
Tabel 3-2. Spesifikasi Robot Movemaster RV-M1	26
Tabel 3-3. Daftar piranti lunak yang digunakan dalam pengembangan sistem	28
Tabel 4-1. Spesifikasi komputer <i>dekstop client</i>	47
Tabel 4-2. Spesifikasi komputer <i>laptop client</i> 1.....	47
Tabel 4-3. Kompatibilitas sistem dengan <i>web browser</i>	48
Tabel 4-4. Waktu pemrosesan masukan hingga menjadi berkas kontrol.....	53



Daftar Gambar

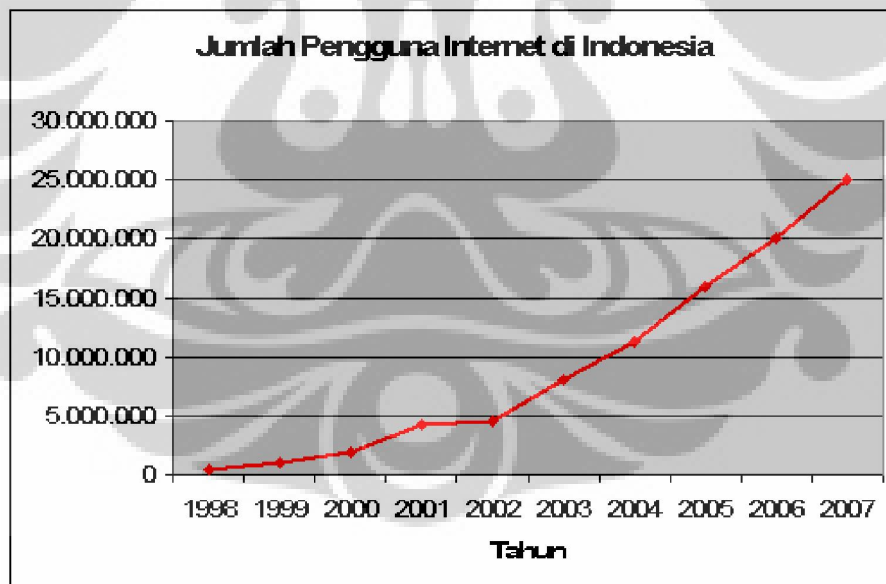
Gambar 1-1. Jumlah pengguna <i>internet</i> di Indonesia tahun 1998 – 2007 [1].....	1
Gambar 2-1. Skema komunikasi mesin-ke-mesin (contoh: komputer-printer).....	5
Gambar 2-2. Skema komunikasi berbasis <i>web</i>	7
Gambar 2-3. Alur pemrosesan dokumen PHP [7].....	11
Gambar 2-4. Contoh tata letak sebuah mikrokontroler Atmega2560.....	15
Gambar 2-5. Desain mekanikal berbagai macam robot [9].....	17
Gambar 2-6. Contoh manipulator berupa sebuah robot artikulasi [10].....	18
Gambar 2-7. Contoh <i>revolute joint</i> [11].....	19
Gambar 2-8. Contoh <i>prismatic joint</i> [12].....	19
Gambar 2-9. Manipulator dengan satu derajat kebebasan [13].....	20
Gambar 3-1. Arsitektur Sistem i-RoMan secara keseluruhan.....	22
Gambar 3-2. Interkoneksi aplikasi di dalam <i>server</i> i-RoMan.....	23
Gambar 3-3. Robot Movemaster RV-M1 [14].....	25
Gambar 3-4. Dimensi Robot Movemaster RV-M1[15].....	26
Gambar 3-5. Area Kerja Robot Movemaster RV-M1[16].....	27
Gambar 3-6. Arah pergerakan tiap <i>joint</i> pada Movemaster RV-M1[17].....	27
Gambar 3-7. Tampilan halaman "Login".....	32
Gambar 3-8. Notifikasi pengguna yang mencoba login saat terdapat sesi pengguna lain yang aktif.....	33
Gambar 3-9. Proteksi dokumen berkas.php.....	33
Gambar 3-10. Halaman Kontrol Utama.....	34
Gambar 3-11. Notifikasi pengguna pada pemrosesan masukan tanpa nilai.....	37
Gambar 3-12. Notifikasi pengguna pada pemrosesan masukan berupa huruf.....	37
Gambar 3-13. Notifikasi pengguna pada pemrosesan masukan di luar rentang kemampuan sistem.....	37
Gambar 3-14. Notifikasi saat akan eksekusi <i>Inverse Kinematics</i>	38
Gambar 3-15. Notifikasi hasil contoh eksekusi <i>Inverse Kinematics</i>	40
Gambar 3-16. Notifikasi saat akan eksekusi <i>Forward Kinematics</i>	41
Gambar 3-17. Notifikasi contoh hasil eksekusi <i>Forward Kinematics</i>	41
Gambar 3-18. Notifikasi saat pengaktifan kondisi darurat.....	42
Gambar 3-19. Notifikasi <i>reset</i> posisi robot.....	43
Gambar 3-20. Posisi <i>default</i> robot [19].....	43
Gambar 3-21. (a) Struktur data pada berkas yang digunakan untuk berkomunikasi antara komputer dengan mikrokontroler. (b) Contoh berkas.....	44
Gambar 3-22. Isi berkas "kdt.txt" untuk (a) Mode darurat. (b) Mode <i>reset</i>	45
Gambar 3-23. Jalur lalu lintas data dari dan ke sistem i-RoMan.....	46
Gambar 4-1. Uji coba mode <i>Inverse Kinematics</i>	50
Gambar 4-2. <i>Dummy data</i> berupa $x = 62$, $y = 112$ dan $z = 207$	51
Gambar 4-3. Uji Coba mode <i>Forward Kinematics</i>	52
Gambar 4-4. Isi berkas kontrol untuk pergerakan <i>Joint</i> 1-3 = -90 dan <i>Joint</i> 4-5 = 90.....	52
Gambar 5. Metode penghitungan waktu pemrosesan.....	53

Bab 1. Pendahuluan

1.1 Latar Belakang Masalah

Tingkat penggunaan teknologi informasi (TI) pada saat ini sudah sangat tinggi dan juga merambah hampir ke segala bidang kehidupan manusia. Tahun 1950-an, ketika komputer pertama kali dikenalkan ke publik dan digunakan secara luas, dunia mengalami kemajuan yang pesat di segala bidang seperti ekonomi dan ilmu pengetahuan.

Internet sebagai salah satu pengembangan dari TI telah dipergunakan secara luas pada masa kini termasuk di Indonesia. Mulai dari sekedar sarana bertukar informasi hingga pada akhirnya terjadi revolusi dunia *internet* yang dikenal dengan sebutan “Web 2.0”, di mana *internet* menjadi lebih interaktif dan didukung tersedianya kecepatan koneksi yang semakin tinggi, aktivitas bisnis dan perdagangan pun dapat dilakukan melalui perantara *internet*.



Gambar 1-1. Jumlah pengguna *internet* di Indonesia tahun 1998 – 2007 [1]

Kecepatan interkoneksi *internet* yang semakin cepat saat ini telah memungkinkan penggunaannya untuk mentransfer data berupa *video* yang berukuran relatif besar, sesuatu yang relatif sulit dilakukan menggunakan internet

pada waktu-waktu yang sebelumnya dikarenakan internet berkecepatan tinggi belum tersedia secara luas. Kemampuan tersebut melahirkan beberapa aplikasi baru yang dikembangkan dengan menggunakan *video* sebagai konten utamanya. Aplikasi tersebut antara lain berupa *video sharing*, *chatting*, hingga *video telephony* yang memungkinkan para pengguna melihat lawan bicaranya masing-masing secara *realtime*.

Faktor-faktor di atas ternyata dapat diaplikasikan untuk mengontrol robot dari jarak jauh melalui media internet. Terutama saat kendala jarak dan waktu menjadi masalah. Terlebih lagi jika yang menjadi kendala adalah medan yang berbahaya bagi manusia. Dengan tersedianya jaringan *internet* yang cepat, maka sebuah robot dapat dipantau pergerakannya secara *realtime* dari mana saja dan kapan saja melalui perangkat yang terkoneksi dengan *internet*.

1.2 Perumusan Masalah

Sebuah sistem dikembangkan untuk mengontrol robot artikulasi dengan lima derajat kebebasan dari jarak jauh melalui *internet* dikarenakan selama ini penggunaannya hanya dapat dioperasikan di tempat (*on-site operation*). Untuk mengatasi masalah tersebut, sistem tersebut harus memenuhi kriteria pengontrolan robot berbasis jaringan *internet* yaitu sistem selain harus memiliki fasilitas yang dapat mewakili panel kontrol yang terdapat pada mesin sebenarnya, juga harus dilengkapi dengan fasilitas pemantauan proses pergerakan secara *realtime* agar pengguna dapat dengan mudah mempergunakan sistem tersebut.

1.3 Pembatasan Masalah

Batasan-batasan yang menjadikan penelitian ini lebih terarah antara lain:

- Robot yang digunakan memiliki 5 (lima) derajat kebebasan (*degree of freedom*) produksi Mitsubishi tipe MovemasterEX RV-M1.
- Pembahasan hingga bagian *user interface* yang berbasis *web* dan komunikasinya antara komputer *client* dengan komputer *server* dengan dua basis pengontrolan yaitu *inverse kinematics* dan *forward kinematics*. Untuk *inverse kinematics* data yang dikeluarkan bersifat *dummy*.
- Tetap menjaga agar komputer pengguna tidak perlu menambahkan program tertentu untuk menjalankan sistem ini. Oleh karena itu, digunakanlah media

internet yang berbasis *web* dan penggunaan *web scripting* yang dipilih adalah HTML, JavaScript dan PHP.

- Piranti keras yang digunakan adalah sebuah komputer yang bertindak sebagai *server* dan dua buah *web camera* serta sebuah *control unit* yang menggunakan *chip* ATmega2560. Untuk piranti lunak, sistem operasi yang digunakan adalah Microsoft Windows XP Service Pack 3. Oleh karena itu, program-program yang digunakan di komputer *server* adalah yang kompatibel dengan sistem operasi tersebut, antara lain penggunaan *web server* Abyss Web Server X1 dan *web camera server* Yawcam.

1.4 Tujuan Penelitian

Penelitian ini bertujuan untuk memungkinkan pengontrolan robot artikulasi secara *off-site* melalui *internet* berbasis *web* sehingga seorang pengguna dapat mengakses dan menjalankan beberapa tugas sederhana dengan robot tersebut dari mana saja dan kapan saja secara *realtime* dan terpantau. Untuk pengujian, sistem dipasang di komputer yang bertindak sebagai *server* yang terhubung dengan jaringan *intranet* Departemen Teknik Mesin Universitas Indonesia (DTM UI) agar bisa diakses melalui komputer lain.

1.5 Metodologi Penelitian

Dalam penelitian ini, metode yang digunakan adalah sebagai berikut:

1. Perumusan masalah yang dihadapi.
2. Perancangan sistem yang disesuaikan untuk mengatasi masalah yang mengemuka dari tahap perumusan masalah.
3. Pembuatan logika sistem, cara kerja/karakteristik sistem dalam menjalankan fungsi yang ditanamkan untuk mengatasi masalah dari tahap perumusan masalah.
4. Pemrograman sintaks *web scripting* (HTML, JavaScript dan PHP) dengan algoritma sesuai dengan logika sistem yang telah dibuat untuk *interface* dan pemrosesan masukan.
5. Uji coba dan analisa sistem yang telah dibuat.
6. Pembuatan laporan

1.6 Sistematika Penulisan

BAB 1. PENDAHULUAN

Pada bab ini dijelaskan mengenai latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan penelitian, metodologi penelitian dan sistematika penulisan.

BAB 2. SISTEM KOMUNIKASI UNTUK KONTROL ROBOT ARTIKULASI BERBASIS *WEB*

Bab ini menjelaskan tentang komunikasi antara komputer *client* dengan komputer *server* yang dibangun khusus untuk mengontrol robot artikulasi yang digunakan dalam penelitian ini. Teori-teori yang dijelaskan mencakup teori-teori seperti pengertian secara mendasar dan cara kerja beberapa piranti keras maupun piranti lunak yang digunakan ataupun dikembangkan guna mendukung penelitian ini.

BAB 3. PENGEMBANGAN SISTEM KOMUNIKASI DAN GUI

Pada bab ini dijelaskan mengenai logika sistem dalam menjalankan fungsinya, spesifikasi sistem yang digunakan untuk membangun keseluruhan sistem, pemrograman *interface* sistem sedemikian rupa sehingga dapat menjalankan fungsi-fungsi sesuai dengan yang ditetapkan di rancangan dan dapat diakses melalui sebuah *web browser* serta implementasi sistem agar dapat diakses dari *internet*.

BAB 4. PENGUJIAN DAN ANALISA SISTEM

Bab ini menjelaskan tentang pengujian dan analisa dari sistem yang telah dibangun termasuk *setup* dari perangkat yang digunakan untuk uji coba. Semua fungsi-fungsi yang telah diprogram diuji coba kemudian hasil yang didapat termasuk anomali sistem yang mungkin terjadi dianalisa.

BAB 5. KESIMPULAN DAN SARAN PENELITIAN LEBIH LANJUT

Bab ini merupakan penutup yang berisi kesimpulan penelitian dan saran untuk penelitian selanjutnya yang lebih maju.

Bab 2. Sistem Komunikasi untuk Kontrol Robot

Artikulasi Berbasis Web

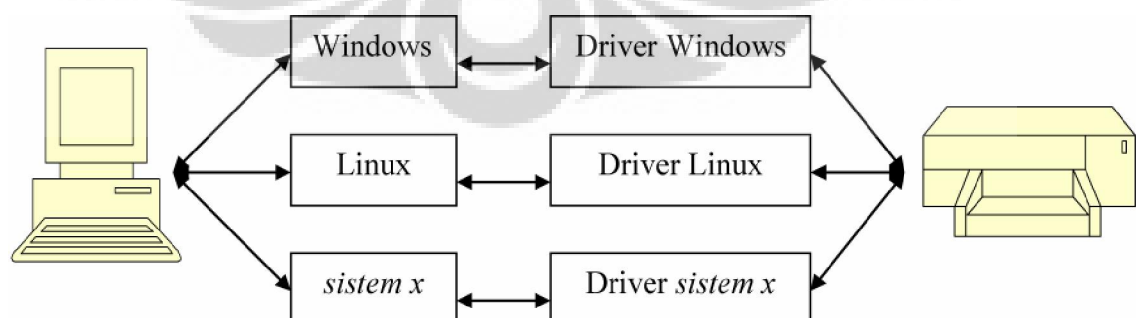
Bab ini menjelaskan tentang komunikasi antara komputer *client* dengan komputer *server* yang dibangun khusus untuk mengontrol robot artikulasi yang digunakan dalam penelitian ini. Teori-teori yang dijelaskan mencakup teori-teori seperti pengertian secara mendasar dan cara kerja beberapa piranti keras maupun piranti lunak yang digunakan ataupun dikembangkan guna mendukung penelitian ini.

2.1 Piranti Lunak

2.1.1 Driver Robot

Robot yang dikontrol secara elektronik dengan mikrokontroler menggunakan bahasa komunikasi yang berbeda dengan bahasa komunikasi yang digunakan oleh sistem operasi seperti Microsoft Windows, Linux dan sebagainya. Oleh karena itu, dibutuhkan suatu kumpulan program yang dinamakan *driver* untuk menerjemahkan bahasa komunikasi yang dikeluarkan oleh sistem operasi menjadi bahasa komunikasi yang dimengerti oleh mikrokontroler sehingga robot dapat dijalankan sesuai dengan yang diperintahkan.

Driver tersebut berbeda-beda, tergantung dari sistem operasi yang digunakan sebagai basis pengontrol robot.



Gambar 2-1. Skema komunikasi mesin-ke-mesin (contoh: komputer-printer)

2.1.2 Web Server

Web server adalah suatu piranti lunak yang berfungsi untuk berkomunikasi dengan *web browser* pengakses yang kemudian memberikan tanggapan berupa kode HTML untuk ditampilkan di *web browser* pengakses.

Selain itu, *web server* juga berfungsi sebagai penerus informasi yang diminta oleh *web browser* ke *interpreter* masing-masing *web scripting* (seperti Perl, PHP, ASP dan sebagainya) apabila dokumen *web* yang diminta oleh pengguna bukan berupa HTML. *Web server* akan memanggil *interpreter* dokumen yang bersangkutan untuk menerjemahkannya kemudian menerima keluaran dari *interpreter* tersebut dan meneruskannya kembali ke *web browser* pengguna. Agar dapat terjalin komunikasi antara *interpreter* dengan *web server*, maka *web server* perlu dikonfigurasi terlebih dahulu.

2.1.3 Web Camera Server

Web camera server adalah suatu piranti lunak yang berperan untuk menangkap gambar yang dihasilkan oleh *web camera* untuk kemudian disiarkan kembali melalui (salah satunya) protokol http. Dengan bantuan piranti lunak inilah, sebuah *web camera* dapat diakses melalui jaringan *intranet/internet* karena piranti ini dapat mengalokasikan *port* tertentu dari sebuah komputer yang terhubung ke jaringan untuk didedikasikan sebagai alamat untuk mengakses *web camera* tersebut.

2.1.4 Web Browser

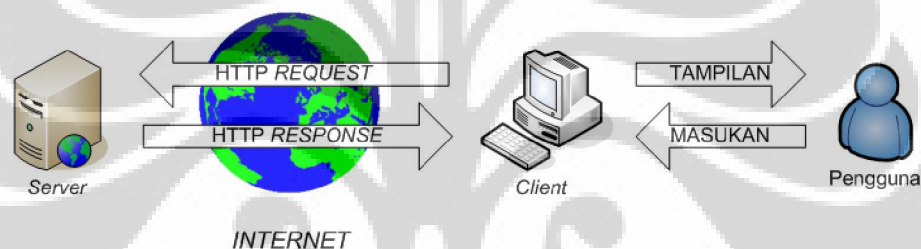
Web browser adalah suatu piranti lunak yang digunakan untuk meminta dokumen kepada *web server* suatu alamat yang dituju, kemudian menampilkannya dalam bentuk terformat (dari dokumen HTML) di layar komputer pengguna. Selain mampu menampilkan, *web browser* dewasa ini juga mampu untuk mengeksekusi *client-side scripting* yang kompatibel dengan HTML yaitu JavaScript.

Namun, metode pengaplikasian JavaScript di tiap-tiap *web browser* cukup berbeda. Sehingga terkadang terdapat ketidakkompatibelan suatu kode JavaScript

ketika di eksekusi di *web browser* satu dengan yang lain. Penjelasan lebih lanjut terdapat di sub-bab *Web Scripting*.

2.2 Protokol Komunikasi HTTP

Dalam komunikasi berbasis *internet*, terdapat berbagai macam protokol untuk berkomunikasi antara suatu komputer dengan komputer lain. Protokol komunikasi HTTP (HyperText Transfer Protocol) adalah satu di antara sekian banyak protokol komunikasi melalui *internet* dalam lingkup *application layer* seperti FTP, IMAP, NTP, SSH dan lain-lain. Protokol HTTP sendiri adalah protokol yang sangat sering dipakai di *web* dan diatur oleh standar RFC 2616 [2]. Cara kerja protokol HTTP sendiri adalah sebagai berikut [3]:



Gambar 2-2. Skema komunikasi berbasis *web*

- Pengguna mengetikkan alamat URL di *web browser*.
- *Web browser* berkomunikasi dengan *name server* untuk menerjemahkan alamat URL yang diketikkan menjadi alamat yang berbentuk numerik. Alamat numerik inilah yang mendeskripsikan alamat suatu dokumen yang dituju sebenarnya.
- *Web browser* kemudian membuat koneksi dan mengirimkan *HTTP request* kepada *web server* dengan alamat yang didapat dari *name server* sebelumnya.
- *Web server* menjawabnya dan mengirimkan *HTTP response* berikut kode HTML dari dokumen yang diminta.

Kode HTML yang diterima selanjutnya di tampilkan dalam bentuk terformat oleh *web browser*.

2.3 Web Scripting

2.3.1 Client-side Scripting

Web scripting jenis ini pertama-tama diunduh terlebih dahulu ke komputer pengguna, setelah itu dikompilasi dan dieksekusi oleh *web browser* dan/atau *plugin* yang terpasang di *web browser* tersebut.

Keuntungan dari *client-side scripting* adalah kecepatan eksekusinya yang tidak tergantung dari kecepatan akses *internet* pengguna. Karena kode dieksekusi di komputer pengguna (setelah kodenya diunduh). Namun, kelemahan dari *client-side scripting* ini adalah kode yang dieksekusi dapat terlihat jelas sehingga bagi sebagian programmer yang menitikberatkan pada masalah hak cipta akan merasa sedikit terganggu. Selain itu, *client-side scripting* tidak memiliki kemampuan untuk menulis berkas ke komputer *server*, namun hal ini semata-mata karena pertimbangan keamanan bagi komputer *server*.

- **HTML [4]**

HTML, bersamaan dengan Hypertext Transport Protocol (HTTP) and Uniform Resource Locator (URL), digagas oleh Tim Berners-Lee pada akhir 1980-an. HTML digunakan olehnya untuk mengorganisir catatan-catatan penelitian sehingga dapat diakses orang lain dan juga mudah ditautkan dengan dokumen lain.

HTML (HyperText Markup Language) adalah bahasa komputer yang terdiri dari sekumpulan *markup* kode dan simbol (biasanya disebut *HTML tag*) yang disisipkan pada suatu file dan digunakan untuk menampilkannya di *web browser*. Kode dan simbol tadi diterjemahkan oleh *web browser* untuk menentukan bagaimana isi dari dokumen tersebut ditampilkan. Khusus untuk *tag* HyperText, *tag* ini membuat isi dari suatu dokumen dapat dihubungkan dengan dokumen lain. *Tag* yang digunakan pada HTML umumnya berpasangan dengan *tag* penutup yang terdapat tambahan “/” pada awal *tag*.

Contoh:

```
<b>Contoh HTML Tag untuk huruf tebal</b>
<i>Contoh HTML Tag untuk huruf miring</i>
<u>Contoh HTML Tag untuk garis bawah</u>
```

ketika *render* oleh *web browser* maka akan menjadi:

Contoh HTML Tag untuk huruf tebal

Contoh HTML Tag untuk huruf miring

Contoh HTML Tag untuk garis bawah

▪ **JavaScript [5]**

Javascript diperkenalkan pertama kali oleh Netscape pada tahun 1995. Pada awalnya bahasa ini dinamakan “LiveScript” yang berfungsi sebagai bahasa sederhana untuk *browser* Netscape Navigator 2. Javascript adalah bahasa yang berbentuk kumpulan *script* yang pada fungsinya berjalan pada suatu dokumen HTML. Sepanjang sejarah *internet*, bahasa ini adalah bahasa *script* pertama untuk *web*. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengizinkan pengekseskuan perintah-perintah di komputer pengguna atau dengan kata lain di sisi *web browser* bukan di sisi *web server*. Javascript sangat bergantung kepada *web browser* yang memanggil halaman *web* yang berisi *script-script* dari Javascript dan tentu saja tersisip di dalam dokumen HTML.

Javascript tidak memerlukan kompilator ataupun penerjemah khusus untuk menjalankannya (pada kenyataannya kompilator Javascript sendiri sudah termasuk di dalam *web browser*). Lain halnya dengan bahasa “Java” (dengan mana Javascript selalu dibanding-bandingkan) yang memerlukan kompilator khusus untuk menerjemahkannya di sisi pengguna.

Contoh:

```
<html>
<body>
<script language="JavaScript" src="tautan ke dokumen JavaScript
jika menggunakan dokumen terpisah">
var data = "Selamat Belajar!";
document.write(d"<b>" + data + "</b>");
</script>
</body>
</html>
```

Dari contoh di atas, pada dasarnya tampil di *web browser* berasal dari kode berikut:

```
<html>
<body>
<b>Selamat Belajar!</b>
</body>
</html>
```

Berhubung JavaScript adalah *client-side scripting*, maka kode di atas hanya terdapat di *memory* komputer pengguna.

2.3.2 Server-side Scripting

Pemrograman *web server-side scripting* tidak melibatkan pengunduhan kode yang dibutuhkan, namun masukan yang diterima langsung diproses oleh dokumen *scripting* di komputer *server* yang diakses. Setelah itu, *server* akan memberikan umpan-balik berupa kode HTML untuk menampilkan hasil pemrosesan tadi kepada pengguna.

Pemrograman *server-side* untuk *web* sangatlah banyak. Mulai dari Python, Perl (cikal-bakal PHP), ASP (Active Server Pages) buatan Microsoft, PHP (PHP Hypertext Preprocessor), Ruby On Rails dan masih banyak lagi.

Dibandingkan dengan *client-side scripting*, *server-side scripting* memiliki beberapa kelebihan yang diantaranya:

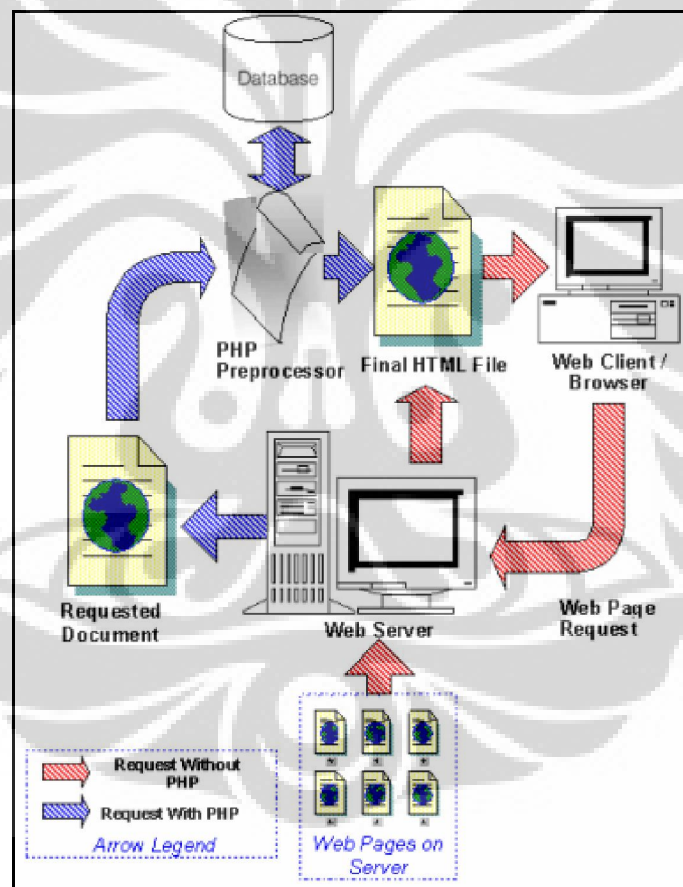
1. Tidak memerlukan tambahan apapun di komputer pengguna untuk mengeksekusi kode yang ada karena kode dieksekusi dan diproses di *server* dan yang dikirim ke pengguna adalah kode HTML yang tinggal ditampilkan oleh *web browser* pengguna.
2. Meskipun sebagian besar bersifat open source, namun pengguna yang mengakses kode *server-side scripting* tidak dapat melihat *source code* dari program yang dijalankan.
3. Memiliki akses untuk membaca dan menulis file ke sistem lokal tempat dokumen *scripting* disimpan dan dieksekusi.

▪ PHP [6]

PHP merupakan nama baru dari PHP/FI yang merupakan cikal-bakal PHP (*PHP Pre Processor*) yang dikembangkan oleh Rasmus Lerdorf pada tahun 1995.

PHP termasuk kategori *server-side scripting* yang ditulis beriringan dengan HTML. Dengan kata lain, sebuah dokumen PHP dapat berisikan kode HTML dan juga kode PHP.

Ketika pengguna mengakses sebuah dokumen PHP, program pemroses PHP – yang dalam dunia pemrograman *web* masuk dalam kategori *interpreter* – menerjemahkan terlebih dahulu kode-kode PHP yang terdapat didalam dokumen tersebut. Kemudian, hasil akhir dari proses interpretasi tersebut disatukan kembali bersama dengan kode HTML yang mungkin terdapat di dalam dokumen PHP tersebut. Terakhir, hasil berupa kode HTML dikirim ke pengguna untuk kemudian ditampilkan di *web browser*.



Gambar 2-3. Alur pemrosesan dokumen PHP [7]

Contoh:

```
<b>Halo, Anonim!</b>
<?php
```

```

echo '<u>';
if ($ _ENV["HTTP_USER_AGENT"] == Mozilla/5.0) {
    echo "Anda menggunakan web browser Mozilla Firefox";
} else {
    echo "Anda tidak menggunakan web browser Mozilla Firefox";
}
echo '</u>';
?>
<i>Semoga hari Anda menyenangkan</i>

```

Contoh kode di atas adalah sebuah dokumen PHP yang juga disisipkan kode HTML. Pada kasus ini, bagian yang diberi warna abu-abu akan terlebih dahulu diproses oleh *interpreter* PHP. Jika seorang pengguna menggunakan *web browser* Mozilla Firefox, maka output yang *web server* berikan ke *web browser* pengguna akan menjadi seperti:

```

<b>Halo, Anonim!</b>
<u>Anda menggunakan web browser Mozilla Firefox</u>
<i>Semoga hari Anda menyenangkan</i>

```

Begitu pula sebaliknya jika *web browser* yang pengguna gunakan bukan merupakan Mozilla Firefox.

2.4 Piranti Keras

2.4.1 Server [8]

Dalam teknologi informasi, sebuah *server* (biasanya disebut *server* aplikasi) adalah “sebuah program aplikasi yang menerima sambungan untuk permohonan layanan dan memberikan kembali tanggapan”. Sebuah *server* aplikasi dapat dijalankan pada komputer yang sama dengan *client* yang menggunakan *server* tersebut, atau dapat tersambung melalui jaringan komputer. Beberapa contoh *server* adalah *file server*, *database server*, *backup server*, *print server*, *mail server*, *web server*, *FTP server*, *application server*, *VPN server*, *DHCP server*, *DNS server*, *WINS server*, *logon server*, *security server*, *domain controller*, *backup domain controller*, *proxy server*, *firewall*, dll.

Dari sisi sistem operasi, *server* merupakan sifat. Sebuah sistem operasi *server* lebih ditujukan untuk menjalankan aplikasi *server*. Perbedaan antara versi sistem operasi *server* dengan *workstation/desktop* umumnya berbeda-beda. Pada

Windows 2000 dan Windows 2000 Server, perbedaan terutama terletak pada jumlah sambungan untuk *network file share*. Pada beberapa edisi sistem operasi *server* biasanya sudah *bundle* dengan aplikasi *server* seperti IIS. Pada Linux hal ini tidak menjadi masalah karena semua sistem operasi Linux dapat dijadikan *server* dengan mudah. Hanya perlu *mentune* dengan benar aplikasi yang kita inginkan dan tidak ada batasan lisensi.

Server komputer (biasanya disebut *server* saja) adalah sistem komputer yang dibuat untuk menjalankan aplikasi *server*. Sebuah komputer *server* yang dialokasikan untuk menjalankan sebuah aplikasi *server* yang spesifik sering kali disebut dengan nama aplikasi tersebut. Contoh, jika *software* Apache HTTP Server dijalankan di komputer *server* perusahaan, maka biasanya di sebut *Web Server* saja. Aplikasi *server* dapat dibagi pada beberapa komputer tergantung pada beban yang ada. Untuk beban yang ringan, semua aplikasi *server* dapat saja dijalankan bersamaan pada sebuah mesin komputer. Untuk beban yang berat, biasanya sebuah aplikasi *server* akan dijalankan pada beberapa mesin komputer sekaligus. Pada beban yang sedang, biasanya sebuah komputer digunakan untuk sebuah aplikasi *server*.

Komputer *server* biasanya diidentifikasi dari model yang dirancang khusus untuk menjalankan aplikasi *server*, biasanya dengan beban tinggi, ditinggalkan, dalam waktu lama mungkin bulanan bahkan tahunan. Sebetulnya semua *workstation* dapat saja menjalankan sistem operasi *server* dan aplikasi *server*, tapi sebuah komputer *server* biasanya mempunyai fitur-fitur yang menyebabkan lebih baik digunakan untuk aplikasi *server*. Beberapa karakteristik sebuah komputer *server* yang membedakannya dengan sebuah *workstation*:

- Memiliki *processor* dan *memory* yang cepat
- Lebih banyak RAM
- *Harddisk* lebih besar
- Lebih *reliable*
- Cadangan *power supply*
- Cadangan *harddisk* (RAID)
- Bentuk dan ukuran yang kompak
- Disain yang modular (seperti *blade server*)

- Dapat dipasang di rak atau kabinet
- Mengarahkan konsol *serial* dll.

Perbedaan utama antara *server* dengan *workstation* sebetulnya bukan pada *hardware*, tetapi lebih kepada *softwarena*. *Server* umumnya menjalankan sistem operasi yang dirancang untuk digunakan di *server*. *Server* akan menjalankan aplikasi yang khusus untuk menjalankan tugas *server*.

Sebuah *server* biasanya memiliki kemampuan *hardware* yang sangat kuat dan sistem yang kompleks. Tidak mengherankan beberapa *server* akan membutuhkan waktu lebih lama untuk *booting* karena harus memeriksa semua peralatan yang tersambung ke *server*. Sering kali, *server* harus melakukan banyak uji coba dan verifikasi sebelum melakukan *booting*. Kadang kali membutuhkan waktu beberapa menit untuk melakukan hal tersebut, sebelum mesin beroperasi selama berbulan-bulan bahkan lebih dari 1-2 tahun lamanya tanpa dimatikan.

Di dunia *desktop*, sistem operasi yang digunakan kebanyakan adalah Microsoft Windows. Sedangkan pada dunia *server*, sistem operasi yang digunakan lebih banyak merupakan turunan sistem operasi UNIX, seperti, FreeBSD, Solaris, dan GNU/Linux. Sistem operasi yang berorientasi untuk *server* umumnya mempunyai beberapa fitur yang mirip yang memang diarahkan untuk keperluan *server*, misalnya:

- Mereka umumnya tidak menggunakan grafik
- Kemampuan untuk mengkonfigurasi ulang hardware atau software tanpa mematikan sistem
- Fasilitas backup yang baik dari data-data yang kritis pada periode tertentu
- Fasilitas untuk *copy* data dari berbagai *volume* atau *device* tanpa mengganggu kerja pengguna
- Kemampuan jaringan yang *advance* dan *flexibel*
- Fasilitas untuk menjalankan *program*/*layanan*/*daemon* tanpa campur tangan operator, Keamanan sistem yang ketat.

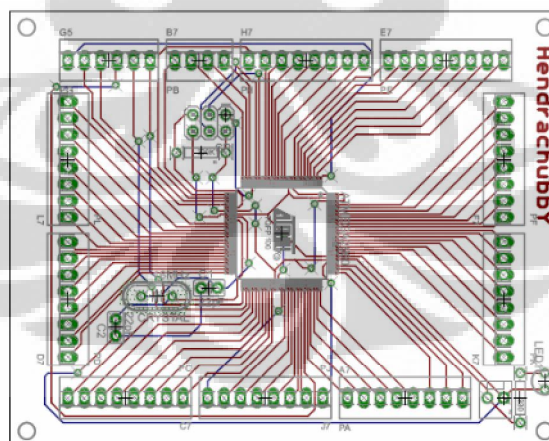
Semakin banyaknya *server* yang berbasis mikroprosesor banyak difasilitasi oleh perkembangan beberapa versi sistem operasi Unix yang berjalan diarsitektur mikroprosesor x86, termasuk Solaris, GNU/Linux dan FreeBSD. Di keluarga

Microsoft Windows, versi Windows NT mulai memasukan fitur yang membuat Windows lebih agak cocok digunakan di *server*.

Walaupun sebetulnya tugas dari sistem operasi *server* dan *desktop* berbeda, dengan kemajuan yang ada pada kinerja dan reliabilitas *hardware* maupun *software*, maka pada hari ini agak sukar untuk membedakan ke dua kelas sistem operasi. Pada hari ini, banyak sistem operasi *desktop* dan *server* menggunakan *source code* yang sama, perbedaan utamanya hanya pada konfigurasi saja. Di samping itu, banyak aplikasi korporasi yang lebih mengacu pada *web based*. Oleh karena itu, *server* aplikasi yang spesifik menjadi tidak terlalu perlu lagi.

Saat ini, hampir semua struktur *internet* berbasis pada model *client-server*. Jutaan *server* tersambung ke *internet* dan berjalan secara terus menerus di seluruh dunia. Ada banyak servis yang diberikan oleh *server* di *internet*, seperti, *web*, Domain Name System (DNS), *e-mail*, *file transfer*, *forum*, *mailing list*, *game online*, *streaming audio* serta *video* dan masih banyak lagi. Pada dasarnya hampir semua hal yang kita lakukan di *internet* akan membutuhkan interaksi dengan *server*.

2.4.2 Control Unit



Gambar 2-4. Contoh tata letak sebuah mikrokontroler Atmega2560

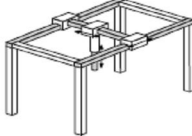
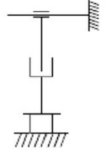
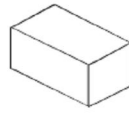

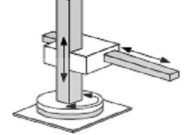
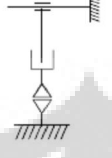


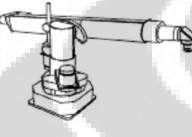
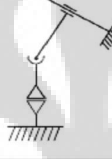


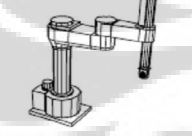
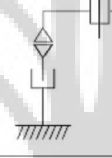


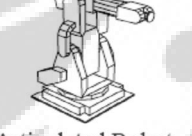
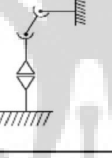



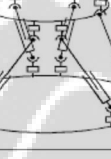


Control unit pada bahasan ini merupakan sebuah piranti keras yang mengontrol robot. Di dalamnya terdapat sebuah mikrokontroler yang merupakan perangkat elektronika yang menyerupai sebuah komputer namun berukuran lebih

kecil dan biasanya berbentuk *integrated circuit* (IC). Layaknya sebuah komputer, mikrokontroler merupakan sebuah *computer processing unit* (CPU) yang memiliki perangkat pelengkap seperti *crystal oscillator*, *timer*, *serial communication*, *digital I/O* dan *analog I/O*. Mikrokontroler menggunakan memory dengan kapasitas kecil yang dapat diprogram ulang dengan komputer melalui bahasa pemrograman yang bervariasi. Mikrokontroler dapat digunakan untuk mengontrol berbagai perangkat elektronik dengan logika pemrograman yang pemrogramannya disesuaikan dengan fungsi alat elektronik tersebut.

Selain mikrokontroler, pada *control unit* terdapat pula *motor driver* yang berfungsi sebagai penggerak *joint* robot. *Motor driver* diperlukan dikarenakan perbedaan tegangan listrik yang dikeluarkan mikrokontroler dengan yang dibutuhkan oleh *motor*. Pada penelitian ini, mikrokontroler menggunakan tegangan 0-5V sedangkan *motor* menggunakan 24V. Selain itu *motor* yang dikontrol juga harus mempunyai kemampuan berbalik arah dan pengaturan kecepatan dijalankan oleh *motor driver*. Sebagai pengaman, *motor driver* juga berguna sebagai pengaman dari arus balik akibat adanya medan elektromagnetis dari *motor*. arus balik ini sangat berbahaya bagi mikrokontroler.

Terakhir adalah *power supply*. Perangkat ini menyediakan arus listrik searah (DC) baik bagi *control unit* maupun bagi motor pada *joint* robot itu sendiri.

2.5 Pengertian Manipulator

Robot	Axes		Examples
	Principle	Kinematic Structure	Workspace
 Cartesian Robot			
 Cylindrical Robot			
 Spherical Robot			
 SCARA Robot			
 Articulated Robot			
 Parallel Robot			

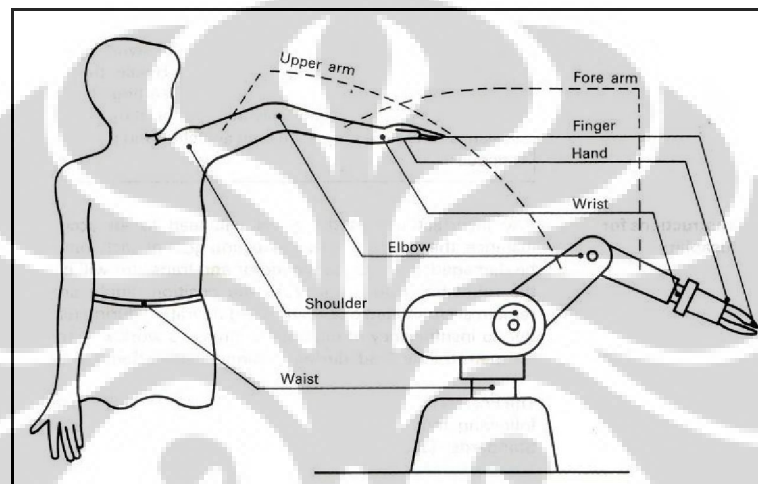
Gambar 2-5. Desain mekanikal berbagai macam robot [9]

Manipulator adalah suatu alat yang dapat bergerak, yang dapat dikontrol oleh penggunanya. Seperti layaknya bagian tubuh manusia, manipulator digunakan oleh manusia untuk membantu pekerjaan-pekerjaan manusia. Secara definisi, manipulator adalah bagian-bagian dan alat-alat yang dapat bergerak di suatu ruangan dengan suatu mekanisme tertentu. Manipulator dapat digunakan untuk membantu manusia mengelas, mengangkat barang, memindahkan barang, dan lain-lain.

Salah satu keunggulan dari manipulator adalah keakuratan. Keakuratan dari suatu manipulator dapat di kontrol oleh manusia. Dan manipulator dapat

melakukan pekerjaan yang berulang-ulang dengan keakuratan yang sama (walaupun untuk itu perlu dilakukan kalibrasi secara berkala).

Suatu manipulator terdiri dari beberapa bagian penyambung yang disambungkan dengan *joint*. *Joint* merupakan suatu penyambung antara kedua *link* yang dapat bergerak dengan arah yang sesuai dengan jenis *joint* tersebut. Terdapat dua jenis *joint* yang biasa digunakan yaitu *Revolute Joint* dan *Prismatic Joint*.

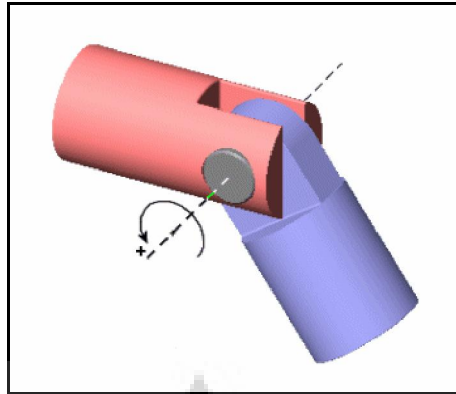


Gambar 2-6. Contoh manipulator berupa sebuah robot artikulasi [10]

Robot tipe artikulasi adalah robot dengan pergerakan yang mirip dengan lengan manusia. Karena kemiripannya, robot-robot tersebut dapat digunakan di pabrik sebagai pengganti pekerja manusia. Robot artikulasi dapat dilengkapi dengan berbagai peralatan pada bagian *end-effector*. Mulai dari *spot welding*, *laser*, *gripper*, hingga *machining tool*. Beberapa robot bahkan telah dilengkapi kecerdasan buatan yang ditanamkan di dalam *controller* robot tersebut sedemikian rupa sehingga robot mampu menganalisa dan merencanakan pergerakan dirinya sendiri tanpa adanya bantuan dari manusia.

2.5.1 Revolute Joint

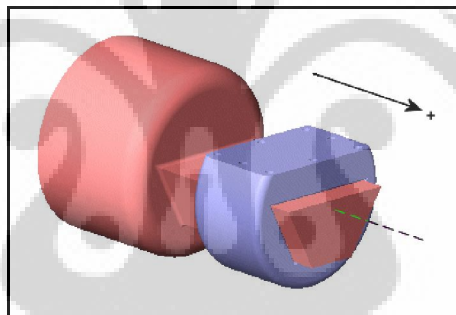
Revolute joint atau *rotary joint* adalah *joint* yang dapat bergerak secara radial. Pada *joint* ini, arah pergerakan hanya searah sumbu radialnya dan tidak dapat bergerak secara aksial. Contoh sederhana penggunaan jenis ini ialah engsel pintu.



Gambar 2-7. Contoh *revolute joint* [11]

2.5.2 Prismatic Joint

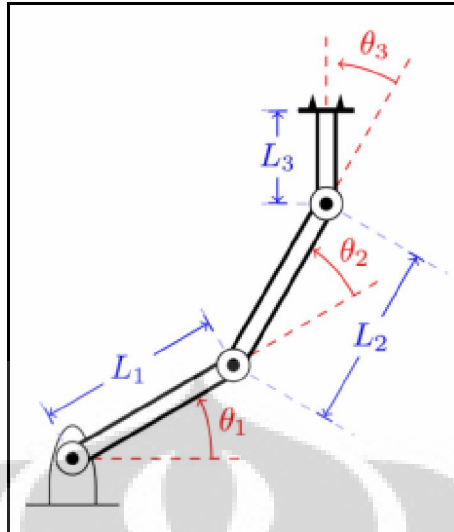
Prismatic joint atau *linear joint* adalah *joint* yang bergerak secara linier. Pada *joint* ini, arah pergerakan hanya sejajar pada sumbu aksialnya.



Gambar 2-8. Contoh *prismatic joint* [12]

2.5.3 Derajat Kebebasan

Di dalam suatu manipulator dikenal istilah derajat kebebasan. Derajat kebebasan adalah suatu perpindahan bebas, baik itu secara aksial ataupun radial, yang perpindahan dari posisi dan orientasi dari suatu manipulator tersebut dapat ditentukan. Suatu manipulator yang memiliki derajat kebebasan yang lebih besar dapat bergerak lebih fleksibel dibandingkan dengan manipulator yang memiliki derajat kebebasan lebih kecil. Semakin besar derajat kebebasan suatu manipulator, pergerakan manipulator tersebut akan semakin fleksibel namun bertambah kompleks.



Gambar 2-9. Manipulator dengan satu derajat kebebasan [13]

Gambar diatas merupakan salah satu contoh manipulator yang memiliki satu derajat kebebasan walaupun manipulator tersebut memiliki tiga *revolute joint*. Banyaknya *joint* yang ada di dalam suatu manipulator belum tentu memberikan tambahan derajat kebebasan dari suatu manipulator. Manipulator pada contoh gambar diatas hanya memiliki satu derajat kebebasan karena manipulator tersebut hanya dapat bergerak dalam satu arah.

Secara alamiah, suatu manipulator memerlukan suatu posisi dan orientasi dari setiap bagian-bagian manipulator tersebut. Untuk dapat menghitung secara matematika jumlah yang menggambarkan suatu posisi dan orientasi dari manipulator, maka dibutuhkan suatu kesepahaman mengenai sistem koordinat. Sistem koordinat yang umum digunakan adalah sistem koordinat kartesius. Setelah sistem koordinat acuan ditetapkan, maka dapat ditentukan deskripsi dari posisi, orientasi dan kerangka kerja dari manipulator tersebut.

2.6 Basis Kontrol Robot

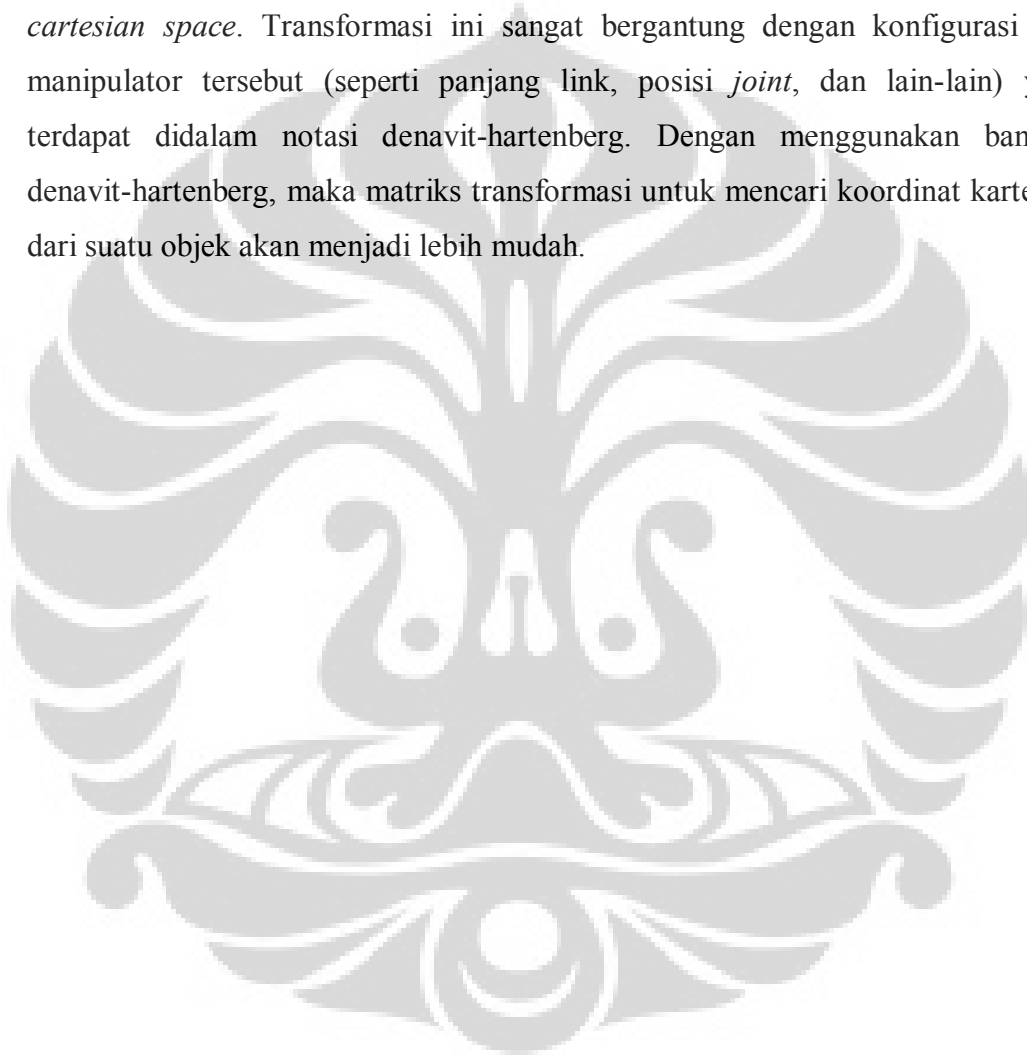
2.6.1 Inverse Kinematics

Inverse kinematics adalah suatu proses untuk mengetahui parameter dari setiap *joint* dari suatu manipulator. Berbeda halnya dengan *forward kinematics*, *inverse kinematics* mentransformasikan dari *cartesian space* ke *joint space*. *Inverse*

kinematics merupakan salah satu aplikasi dalam robotika yang mencari tahu parameter dari manipulator untuk mendapatkan posisi yang telah ditentukan.

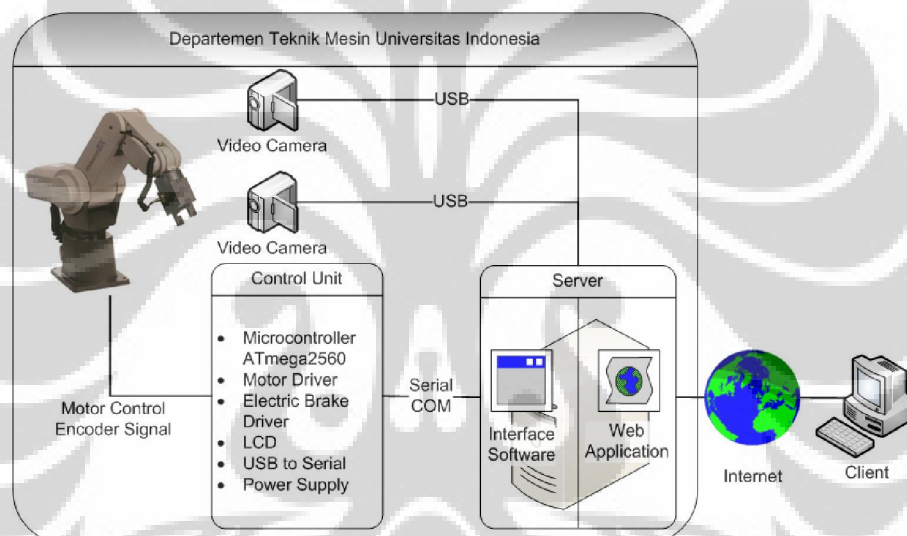
2.6.2 Forward Kinematics

Forward kinematics adalah suatu cara menggambarkan posisi dari suatu manipulator dengan mengetahui pergerakan dari setiap *joint* manipulator. *Forward kinematics* merupakan suatu transformasi dari *joint space* kedalam *cartesian space*. Transformasi ini sangat bergantung dengan konfigurasi dari manipulator tersebut (seperti panjang link, posisi *joint*, dan lain-lain) yang terdapat didalam notasi denavit-hartenberg. Dengan menggunakan bantuan denavit-hartenberg, maka matriks transformasi untuk mencari koordinat kartesian dari suatu objek akan menjadi lebih mudah.



Bab 3. Pengembangan Sistem Komunikasi dan GUI

Pada bab ini dijelaskan mengenai logika sistem yang diberi nama “i-RoMan” (kependekan dari “*internet Robot Manipulation*”) dalam menjalankan fungsinya. Spesifikasi sistem yang digunakan untuk membangun keseluruhan sistem, pemrograman *interface* sistem sedemikian rupa sehingga dapat menjalankan fungsi-fungsi sesuai dengan yang ditetapkan di rancangan dan dapat diakses melalui sebuah *web browser* serta implementasi sistem agar dapat diakses dari *internet*.

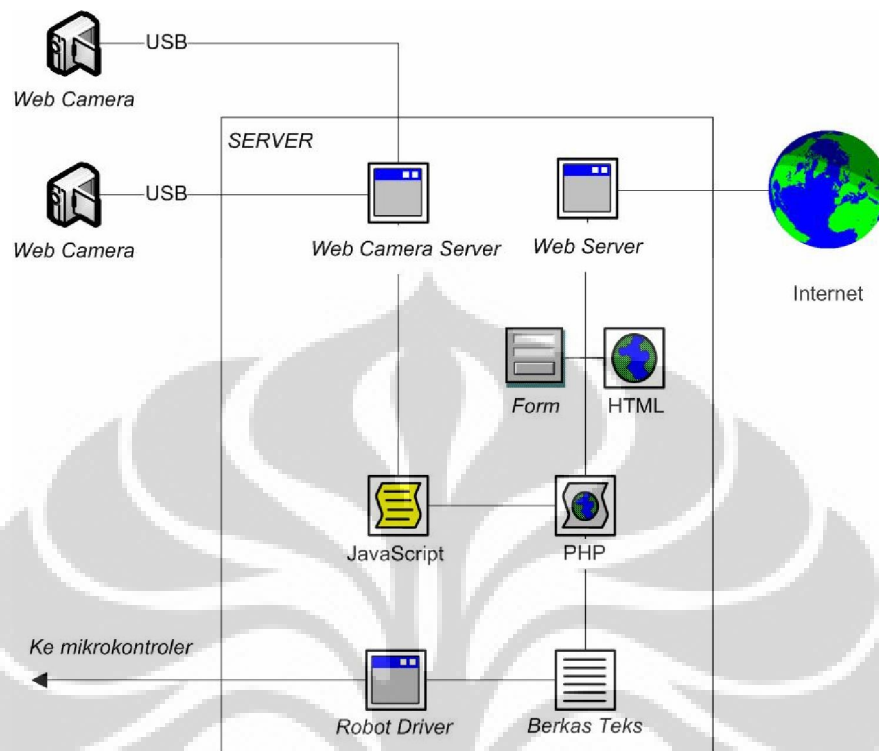


Gambar 3-1. Arsitektur Sistem i-RoMan secara keseluruhan

Sistem i-RoMan terdiri dari beberapa piranti keras maupun piranti lunak baik yang terpasang pada komputer *server* maupun terhubung secara langsung ataupun tidak langsung ke komputer *server*. Untuk piranti lunak, sebuah *web server*, *web camera server* dan *driver robot* terpasang di komputer *server*. Sedangkan untuk piranti keras, komputer *server* terhubung dengan dua buah *web camera* dan sebuah mikrokontroler yang berfungsi sebagai pemantau dan pengontrol robot.

Sesuai dengan pembatasan masalah, pembahasan hanya sebatas sistem komunikasi dan *interface* untuk kontrol robot artikulasi berbasis aplikasi yang dikembangkan untuk *web*. Aplikasi-aplikasi tersebut diletakkan dan dijalankan di

sebuah komputer *server* yang terletak di Laboratorium Manufaktur dan Otomasi DTM UI.



Gambar 3-2. Interkoneksi aplikasi di dalam server i-RoMan

3.1 Logika Sistem

Logika dari sistem ini ialah memindahkan panel kontrol robot ke dalam suatu *interface* berbasis *web*. Sistem ini mengizinkan pengontrolan robot dari jarak jauh melalui aplikasi *web* berbasis jaringan *internet*. Selain itu, kondisi robot yang dikontrol dapat dipantau secara *realtime* menggunakan *web camera* sehingga pengguna dapat mengetahui status dan kondisi robot setiap saat termasuk memantau reaksi dari perintah yang diberikan pengguna.

Untuk mengontrol robot, di sistem ini terdapat dua mode yaitu berbasis *Inverse Kinematics* dan *Forward Kinematics*. Pemilihan mode ditentukan berdasarkan masukan yang diberikan oleh pengguna dari halaman kontrol.

3.1.1 Inverse Kinematics

Pada halaman kontrol nantinya terdapat dua gambar yang digunakan untuk memantau posisi dan pergerakan robot. Pengguna dapat secara langsung mengklik

gambar tersebut – pada gambar **Cam_1** untuk koordinat pada bidang x dan y serta gambar **Cam_2** untuk koordinat pada bidang z – sesuai posisi yang diinginkan untuk robot untuk bergerak.

Pertama kali sistem diakses, robot selalu dalam kondisi *default*. Setelah itu pengguna mengklik gambar *web camera* yang menampilkan status dan kondisi robot secara *realtime* untuk mendapatkan koordinat posisi yang diinginkan. Setelah pengguna mengkonfirmasi koordinat yang dimasukkan, sistem kemudian akan memroses dan menggerakkan robot sesuai dengan data koordinat yang dimasukkan.

3.1.2 Forward Kinematics

Pada mode ini, pengguna langsung memasukkan nilai untuk pergerakan tiap-tiap nilai *joint* berupa sudut dalam derajat secara manual. Nilai positif dan negatif dari masukan yang diterima akan dikenali oleh sistem sebagai arah. Nilai positif untuk arah ke atas/kanan dan nilai negatif untuk arah ke bawah/kiri, tergantung dari pergerakan tiap-tiap *joint* pada robot.

Pada mode ini, selain sebagai berfungsi sebagai pemantau, gambar dari *web camera* tersebut juga digunakan untuk memperkirakan apakah data yang dimasukkan dapat dieksekusi oleh sistem (misalnya apakah robot akan menyentuh *limit switch* pada *joint* tertentu atau menabrak benda lain).

3.2 Spesifikasi Sistem

3.2.1 Piranti Keras

- **Server**

Sistem *server* merupakan satu kesatuan dari *web server*, *web camera server* dan juga *interpreter* PHP. Ketiganya diintegrasikan melalui bahasa HTML untuk mengatur tata letak sehingga didapat suatu sistem yang terintegrasi yang dapat melakukan banyak tugas sesuai yang telah dirancang dalam satu *platform*.

Sistem *server* ini terhubung ke jaringan *internet* melalui *intranet* DTM UI.

Tabel 3-1. Spesifikasi komputer *server*

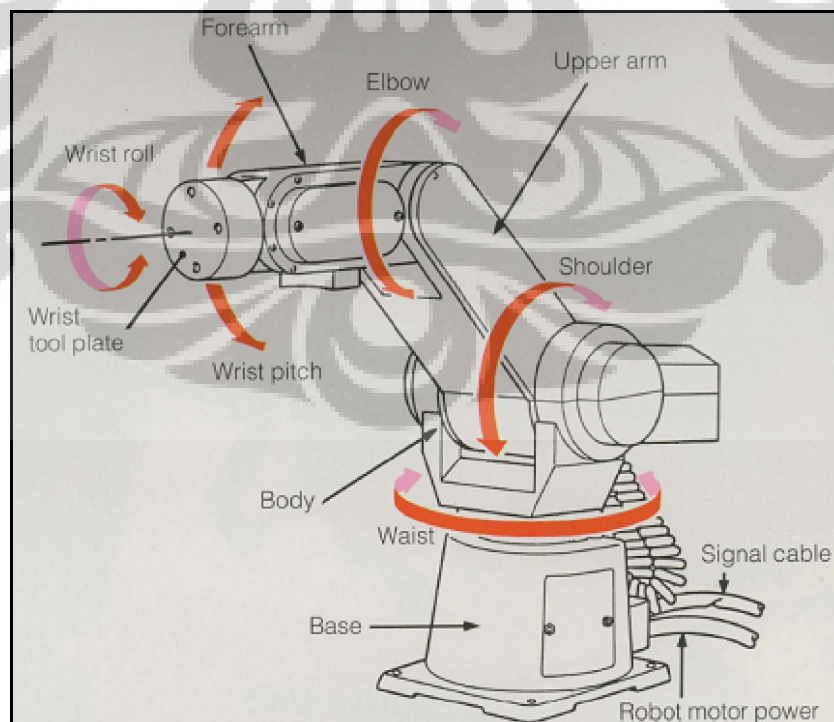
Jenis	Nama
-------	------

<i>Prosesor</i>	Intel Pentium D Dual-core 3 GHz
<i>RAM</i>	1 GB DDR2-SDRAM
<i>HDD</i>	80 GB
<i>Web Camera</i>	2 x Logitech QuickCam E3500 Plus
<i>Router Jaringan</i>	D-Link 10/100 DES-1008D
<i>I/O Connection</i>	USB (Universal Serial Bus)

Web camera yang digunakan di komputer server memiliki resolusi *default* sebesar 640x480 piksel. Namun, yang digunakan oleh sistem hanya sampai dengan resolusi 320x480. Hal ini semata-mata untuk mengecilkan ukuran berkas gambar tiap *frame* yang harus dikirim oleh *server* ke komputer pengguna.

Lebih lanjut lagi, tiap berkas gambar yang akan dikirim ke pengguna nantinya dikompresi terlebih dahulu hingga 25% dari ukuran aslinya. Sehingga ukuran berkas gambar akan menjadi lebih kecil lagi namun tetap dalam kualitas yang masih memungkinkan untuk pengoperasian sistem (dapat dibedakan antara robot dengan lingkungan sekitarnya).

- **Robot**

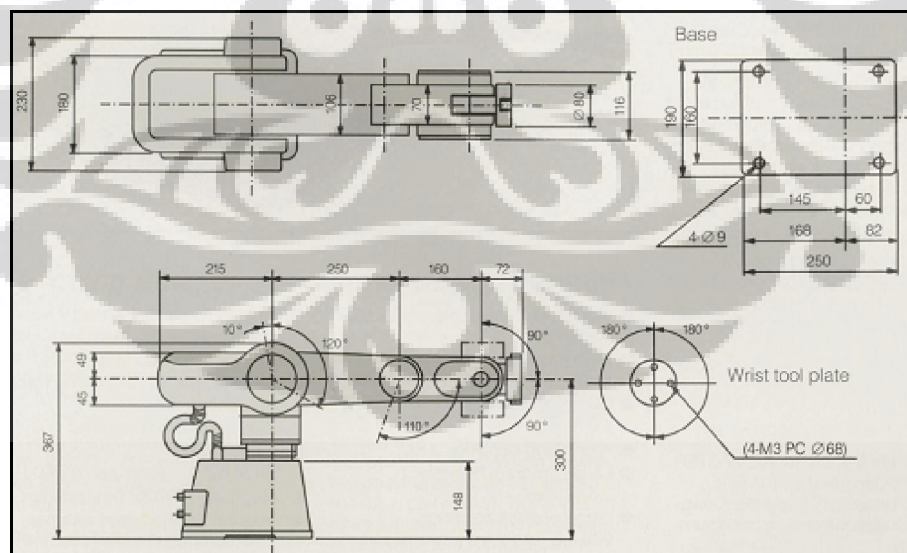


Gambar 3-3. Robot Movemaster RV-M1 [10]

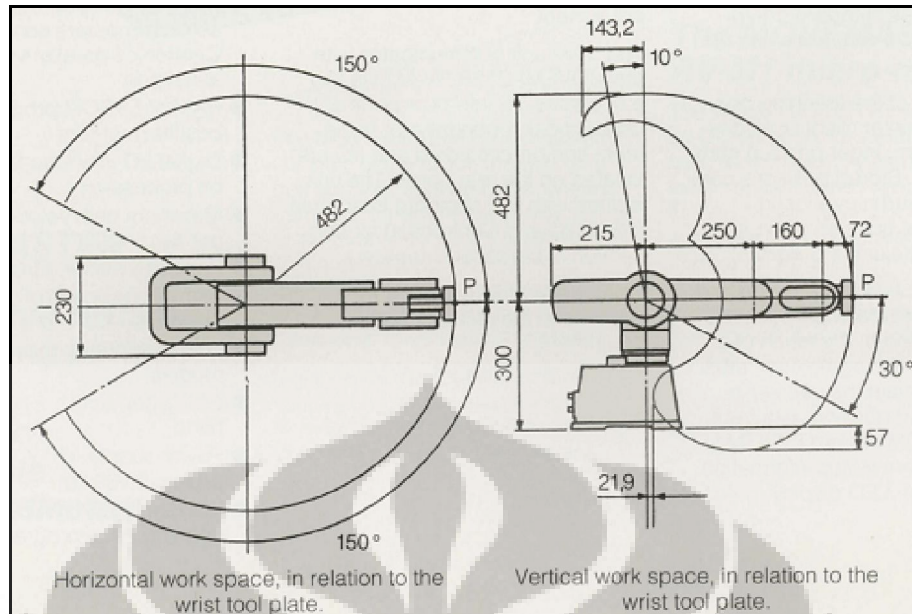
Robot yang digunakan dalam penelitian ini adalah Movemaster RV-M1 produksi dari perusahaan Mitsubishi dengan spesifikasi sebagai berikut:

Tabel 3-2. Spesifikasi Robot Movemaster RV-M1 [10]

	Keterangan
<i>Konstruksi</i>	Vertikal
<i>Derajat Kebebasan</i>	5 (lima)
<i>Sistem Penggerak Elektrik</i>	Motor DC
<i>Jangkauan</i>	250 + 160mm
<i>Kecepatan Maksimum</i>	1000mm/s
<i>Kapasitas Lifting</i>	1.2Kg
<i>Repeatability</i>	± 0.3mm
<i>Penunjuk Arah</i>	Limit Switch & Encoders
<i>Posisi Pemasangan</i>	Horizontal
<i>Temperatur Ambient</i>	15C - 40C
<i>Berat</i>	19Kg



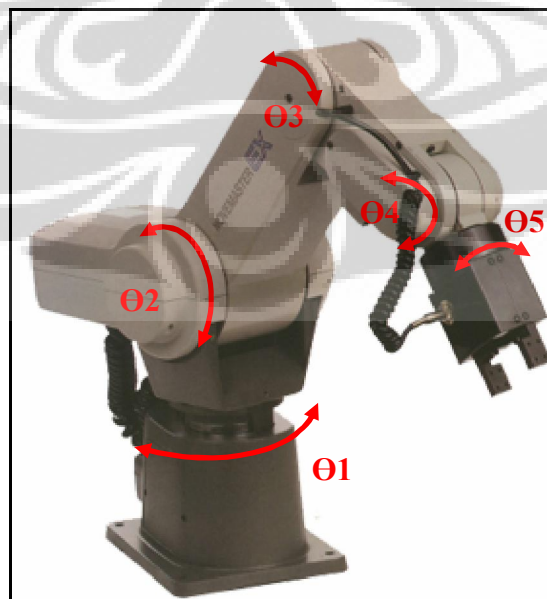
Gambar 3-4. Dimensi Robot Movemaster RV-M1[10]



Gambar 3-5. Area Kerja Robot Movemaster RV-M1[10]

Sistem ini membagi *joint* robot Movemaster RV-M1 menjadi lima:

1. Joint 1: Pergerakan robot ke arah kiri-kanan
2. Joint 2: Pergerakan robot ke arah atas-bawah
3. Joint 3: Pergerakan robot ke arah atas-bawah
4. Joint 4: Pergerakan robot ke arah atas-bawah
5. Joint 5: Pergerakan robot ke arah kiri-kanan



Gambar 3-6. Arah pergerakan tiap *joint* pada Movemaster RV-M1[14]

3.2.2 Piranti Lunak

Tabel 3-3. Daftar piranti lunak yang digunakan dalam pengembangan sistem

Jenis	Nama
Sistem Operasi	Microsoft WindowsXP Service Pack 3
<i>Web Server</i>	Abyss Web Server X1 version 2.6
<i>Web Camera Server</i>	Yawcam
Piranti Pengembangan	PHP 5.2.3, Notepad++
<i>Web Browser</i>	Mozilla Firefox 3.5.x dan Internet Explorer 6.0-8.0

Untuk sistem operasi, digunakan solusi dari Microsoft yaitu Microsoft WindowsXP dengan Service Pack 3. Untuk *web server* sendiri, dipilih solusi dari Aprelium yaitu Abyss Web Server X1 versi 2.6. *Web camera server* dipilih solusi dari Yawcam sedangkan untuk pemrograman digunakan *interpreter* PHP versi 5.2.3 dan *editor* teks Notepad++. Untuk pengujian, digunakan *web browser* Mozilla Firefox versi 3.5.5 dan *Internet Explorer* versi 6.0, versi 7.0 dan 8.0. Kecuali sistem operasi, piranti lunak lainnya berlisensi *freeware* atau dengan kata lain dapat dipergunakan secara cuma-cuma.

Untuk menggerakkan robot, dikembangkan pula suatu piranti lunak bernama “DriverRV” yang berfungsi sebagai *driver* robot untuk berkomunikasi antara komputer *server* dengan mikrokontroler pengontrol robot. Piranti lunak ini sendiri dikembangkan dengan bahasa pemrograman C.

3.3 Web Scripting

Sistem dikembangkan dengan *web scripting* yang menggunakan tiga bahasa *web scripting* yaitu HTML, JavaScript dan PHP. Banyaknya bahasa yang digunakan diakibatkan oleh kompleksnya cara kerja sistem dan keterbatasan kemampuan masing-masing bahasa pemrograman serta penyesuaian terhadap kondisi standar (bawaan) sebuah komputer yang telah disebutkan sebelumnya bahwa sistem ini akan dapat bekerja tanpa tambahan apapun pada komputer pengguna.

Hal ini ditambah dengan kenyataan bahwa bahasa pemrograman JavaScript dan PHP termasuk kategori *high-level*, yaitu banyak fungsi yang sudah tertanam didalam intinya. Sehingga untuk membuat suatu fungsi tertentu, seorang *programmer* cukup memanggilnya saja dan tidak perlu memrogramnya sendiri.

- **HTML**

Sebagai bahasa standar bagi *web browser* untuk menampilkan elemen-elemen *web*, penggunaan bahasa HTML tidak dapat mungkin dihindari lagi. Sistem ini dikembangkan dengan bahasa HTML pada bagian *user interface*. Hal tersebut meliputi tampilan latar belakang, formulir pengisian, tampilan gambar dan tata letak. Kesemuanya dibangun di atas bahasa HTML yang kemudian ditampilkan secara *formatted* oleh *web browser*.

- **JavaScript**

Bahasa pemrograman JavaScript merupakan salah satu inti utama sistem ini. Tampilan langsung dari *web camera* yang diperbarui empat kali setiap detik, peringatan-peringatan dan konfirmasi masukan pada saat pengisian formulir hingga penanganan kesalahan pengisian dan penanganan kondisi darurat diprogram dengan bahasa JavaScript. Pertimbangan penggunaan bahasa pemrograman JavaScript sendiri didasarkan atas kecepatan eksekusi yang relatif lebih cepat dibanding bahasa pemrograman *web* lainnya karena yang mengeksekusi adalah komputer *client*, tidak melibatkan komputer *server* yang melibatkan proses kirim-terima data yang akan diproses antar komputer.

- **PHP**

Alasan utama penggunaan PHP dalam pengembangan sistem ini adalah karena *client-side scripting* seperti JavaScript tidak mampu/tidak diperbolehkan menulis apapun ke komputer tempat kode tersebut berada. Selain itu pemrosesan formulir masukan dilakukan oleh PHP karena ketidakmampuan JavaScript untuk melakukannya, sehingga terdapat pembagian beban kerja antara komputer pengguna dengan komputer *server*. Alasan lain penggunaan PHP adalah karena PHP mampu mengakses variabel-variabel baik yang terdapat di komputer *client* ataupun *server*. Penggunaan tag “<?php” dan “?” dalam suatu dokumen PHP untuk memberi tahu *interpreter* PHP bagian mana yang harus diproses sangat membantu pada fase pengembangan sehingga dokumen HTML dapat langsung

disisipkan ke dalam dokumen PHP. Hal ini sangat mengurangi kerumitan pemrograman.

3.4 Interface

Bagian *interface* sistem ini dirancang terdiri dari:

- a. Manajemen Sesi dan Proteksi
- b. Halaman Kontrol Utama
- c. Pengambilan Gambar *Web Camera* dan Koordinat *Cursor Mouse*
- d. Penanganan Kesalahan

Algoritma pada bagian *interface* ini secara keseluruhan adalah sebagai berikut:

```
//Halaman Otentikasi
If kecepatan koneksi memenuhi
    Tampilkan halaman otentikasi
    If password benar
        Masuk halaman kontrol utama
    Else
        Notifikasi pengguna
        Blokir akses
Else
    Notifikasi pengguna
    Blokir akses

//Halaman kontrol utama
If logout counter = 0
    Logout otomatis
If masukan valid
    Konfirmasi ke pengguna
    If konfirmasi OK
        Proses
        Notifikasi masukan yang diterima
    Else
        Kembali
Else
    Notifikasi pengguna sesuai kesalahan
```

3.4.1 Manajemen Sesi dan Proteksi

Sistem ini dikembangkan agar pengguna dapat mengontrol robot melalui *internet*. Untuk mendukung fungsi tersebut dibutuhkan beberapa mekanisme untuk memproteksi, baik sistem maupun pengguna itu sendiri. Salah satunya adalah pengecekan kecepatan akses *internet* pengguna. Agar gambar yang ditangkap oleh *web camera* dapat ditampilkan dengan baik di komputer pengguna, sistem akan mengukur terlebih dahulu kecepatan akses *internet* pengguna apakah

memenuhi batasan minimum yang ditetapkan. Berikut adalah perhitungan untuk menentukan kecepatan minimum akses *internet*.

Jumlah Layar Web Camera = 2

Jumlah fps per layar = 4

Ukuran gambar per satu frame » 20 kilobyte (maks.)

Maka kecepatan minimum = $2 \times 4 \times 20$

= 160 kilobyte/detik

1 byte = 8 bit, maka: $160 \times 8 = 1280$ kilobit/detik

dengan asumsi perubahan kecepatan akses *internet* $\pm 5\%$, maka:

kecepatan minimum = kecepatan minimum + ($5\% \times$ kecepatan minimum)

= $1280 + (1280 \times 0,05)$

= 1344 Kilobit/detik = 168 kilobyte/detik

Perhitungan di atas valid untuk lokasi sistem di DTM UI. Ukuran dokumen gambar sangat tergantung dari tipe *web camera*, resolusi gambar yang digunakan dan juga cahaya serta benda-benda di sekitar robot yang masuk ke dalam *frame*.

Selain itu sistem juga diproteksi dari akses oleh dua pengguna secara bersamaan dan juga akses ke dokumen manipulator oleh pengguna yang tidak memiliki sesi aktif. Khusus untuk halaman kontrol utama, bagian ini diproteksi oleh mekanisme otentikasi dan juga sesi. Sehingga hanya pengguna yang memiliki password dan sesinya masih aktif saja yang dapat mengakses sistem.

Algoritma Program:

```
//Pengecekan Kecepatan
If kecepatan akses internet pengguna < 1344 Kbps (168 KBps)
    Blokir akses ke sistem dan notifikasi pengguna

//MANAJEMEN SESI
//Login
If nilai berkas kontrol sama dengan 0 (nol)
    Tampilkan halaman login
    If password benar
        Enkripsi password
        Buat cookies berisi password yang terenkripsi
        Catat IP pengguna ke berkas kontrol
        Masuk halaman kontrol utama
Else if nilai berkas kontrol tidak sama dengan 0 (nol)
    If nilai berkas kontrol sama dengan IP pengguna
        If cookies ada dan password benar
            Lanjutkan sesi
Else
    Blokir akses ke sistem dan notifikasi pengguna
```

```
//Logout
If nilai berkas kontrol sama dengan IP pengguna
  If cookies ada
    Hapus cookies
    Set nilai berkas kontrol ke 0 (nol)
Else blokir akses dan notifikasi pengguna
```

Gambar 3-7. Tampilan halaman "Login"

Setelah login berhasil, akan tercipta suatu sesi aktif dari pengguna sehingga ketika ada pengguna lain yang berusaha mengakses sistem, sistem akan menolak dan memberitahu bahwa terdapat pengguna lain yang sedang mengakses sistem tersebut. Algoritma di atas akan selalu dipanggil setiap kali halaman kontrol utama (dokumen "index.php") dan halaman pemrosesan masukan (dokumen "berkas.php") dipanggil/dieksekusi sistem.

Dari algoritma di atas, terlihat bahwa untuk memproteksi halaman kontrol utama terdapat banyak mekanisme. Pertama adalah pemeriksaan pengguna aktif yaitu apabila terdapat pengguna lain yang sedang aktif, maka pengguna berikutnya yang mencoba untuk mengakses sistem akan diblokir secara otomatis.



Gambar 3-8. Notifikasi pengguna yang mencoba login saat terdapat sesi pengguna lain yang aktif

Berikutnya apabila ternyata berkas kontrol tidak 0 (no1), maka sistem akan mencocokkan IP pengguna dengan catatan yang dimiliki sistem di berkas kontrol. Apabila cocok, maka pengguna tersebut dapat melanjutkan sesinya, apabila tidak cocok, maka pengguna akan diblokir. Hal ini untuk mencegah pengguna yang membuka halaman kontrol utama di *window* atau *tab* yang baru. Sementara sesi di *window* atau *tab* yang lama telah berakhir.



Gambar 3-9. Proteksi dokumen berkas.php

Pada **Gambar 3-9** dapat dilihat mekanisme proteksi yang dikembangkan untuk mencegah akses dokumen “berkas.php” yang berfungsi untuk menulis berkas kontrol gerakan robot, secara langsung.

3.4.2 Halaman Kontrol Utama



Gambar 3-10. Halaman Kontrol Utama

Algoritma Program:

```
//Pembaruan Sesi
If melakukan pemrosesan atau mengaktifkan kondisi darurat
  Reset counter
  Reset kadaluarsa cookies
Else if idle >9 menit
  Logout otomatis

//Pencegah Penutupan Windows
If window halaman kontrol ditutup
  Konfirmasi ke pengguna
  If jawaban "ya"
    Tutup halaman kontrol
    Logout paksa pengguna
    Hapus cookies kontrol

//Pemilihan Mode Kontrol
If nilai ID konfirmasi sama dengan 1
  Kirim nilai 1 ke algoritma pemrosesan
  Mulai proses inverse kinematics
Else if nilai ID konfirmasi sama dengan 2
  Kirim nilai 1 ke algoritma pemrosesan
  Mulai proses forward kinematics

//Monitor event
If tombol darurat ditekan atau data masukan diproses sistem
  Reset logout counter ke nilai awal
  Perbarui kadaluarsa cookies
```

```

Else if tombol "Esc" ditekan
    Masuk mode darurat
Else if nilai logout counter sama dengan "0:0"
    Logout pengguna secara otomatis

```

Pada halaman kontrol utama, terdapat semacam mekanisme proteksi dari tidak sengajanya *window* halaman kontrol utama ditutup oleh pengguna. Selain itu, sistem juga dirancang untuk logout secara otomatis apabila tidak ada kegiatan atau proses apapun yang dilakukan oleh pengguna selama sembilan menit. Namun, apabila dalam rentang waktu sembilan menit tersebut pengguna tetap melakukan proses (eksekusi perintah dan/atau mengaktifkan kondisi darurat dan/atau mereset posisi robot), maka *counter* logout akan direset dan cookies akan diperbarui kadaluarsanya.

3.4.3 Pengambilan Gambar Web Camera dan Koordinat Cursor Mouse [15]

Algoritma Program:

```

//Pengambilan Gambar dari web camera
Refresh 4 (empat) kali setiap detik
Perbarui gambar di "Cam 1" dan "Cam 2"

//Pergerakan cursor mouse
If pointer mouse bergerak di atas imej web camera
    If di imej kiri
        tangkap koordinat pointer mouse relatif terhadap imej
        tangkap koordinat mouse kemudian kirim ke form "sumbu
x,y"
    Else if di imej kanan
        tangkap koordinat pointer mouse relatif terhadap imej
        tangkap koordinat mouse kemudian kirim ke form "sumbu z"

//Klik mouse
If tombol mouse kiri diklik di imej kiri
    tangkap koordinat mouse kemudian kirim ke form "sumbu x,y"
Else If tombol mouse kiri diklik di imej kanan
    tangkap koordinat mouse kemudian kirim ke form "sumbu z"

```

Untuk mendukung pengontrolan robot secara *realtime*, khususnya mode kontrol robot *inverse kinematics*, dirancanglah sebuah metode pengambilan gambar yang ditangkap oleh *web camera* melalui *web camera server* dan juga metode pengambilan koordinat *cursor mouse* dari gambar tersebut. Berdasarkan kesepakatan tim pengembang, maka ditetapkan bahwa *refresh rate* dari *web*

camera adalah empat *frame*/detik atau dengan kata lain gambar kondisi dan pergerakan robot akan diperbarui empat kali setiap detiknya.

Sistem akan menangkap koordinat *cursor mouse* selama *cursor mouse* tersebut berada dan/atau bergerak diatas gambar *web camera*. Koordinat yang ditampilkan adalah relatif terhadap titik paling kiri atas gambar.

Khusus untuk mode kontrol robot secara *inverse kinematics*, koordinat ini dapat digunakan sebagai masukan bagi sistem untuk menggerakkan robot ke koordinat tertentu pada gambar. Caranya adalah dengan mengklik pada titik yang diinginkan, maka koordinat *cursor mouse* yang ditangkap oleh sistem akan dimasukkan ke *form* pemrosesan.

3.4.4 Penanganan Kesalahan

Karena proses memasukkan nilai-nilai yang ada menggunakan masukan melalui alat berupa *keyboard* komputer, maka sangat mungkin bagi pengguna untuk salah memasukkan masukan yang valid. Untuk itu, sistem ini didesain untuk menangani kesalahan-kesalahan yaitu:

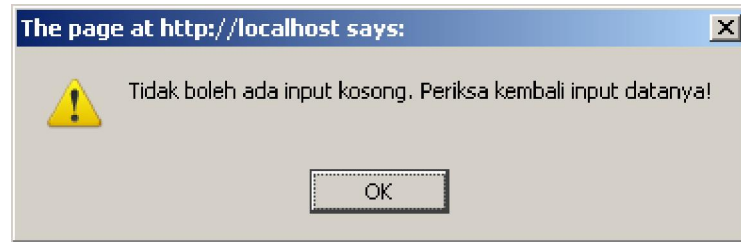
- Memroses masukan tanpa nilai
- Masukan bukan berupa angka
- Masukan di luar jangkauan sistem

Algoritma Program:

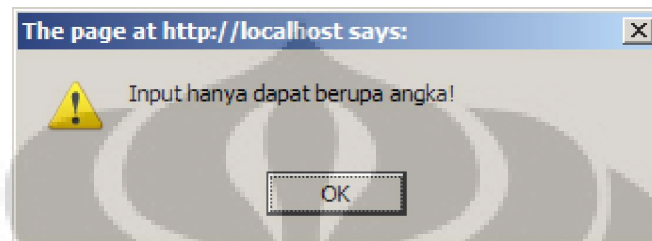
```
//Validasi masukan kosong
If form diproses
  If form tidak memiliki masukan
    Notifikasi pengguna

//Validasi angka
If masukan pada form bukan angka
  Hapus nilai
  Ubah background form menjadi merah
  Notifikasi pengguna
Else if masukan pada form berupa angka
  If nilainya di luar rentang sistem
    Notifikasi pengguna

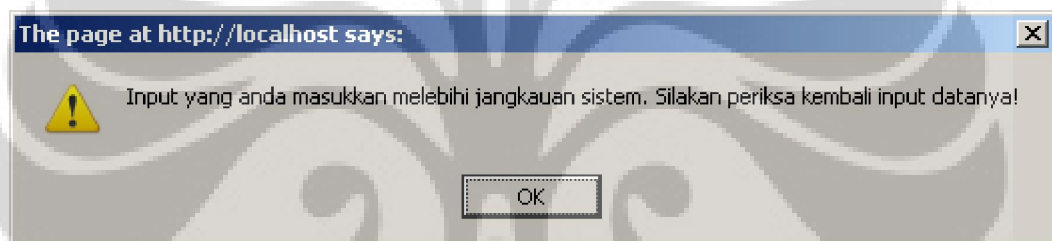
//Validasi rentang masukan
If masukan ">" atau "<" dari rentang yang telah ditentukan
  Notifikasi pengguna
```



Gambar 3-11. Notifikasi pengguna pada pemrosesan masukan tanpa nilai



Gambar 3-12. Notifikasi pengguna pada pemrosesan masukan berupa huruf



Gambar 3-13. Notifikasi pengguna pada pemrosesan masukan di luar rentang kemampuan sistem

3.5 Bagian pemrosesan

Bagian pemrosesan merupakan bagian yang penting dari sistem ini. Di sini, masukan yang diterima ditentukan modennya, apakah *inverse kinematics* atau *forward kinematics*. Selain memroses data pergerakan robot, bagian pemrosesan juga memroses masukan untuk mengaktifkan kondisi darurat dan juga masukan untuk mereset posisi robot ke posisi *default*. Posisi *default* robot dapat dilihat pada Gambar 3-20.

3.5.1 Pemrosesan Masukan

- **Mode *Inverse Kinematics***

Algoritma Program:

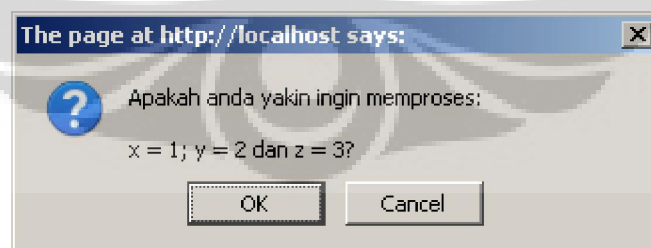

```

If ID pemrosesan sama dengan 1
  //Mode Inverse Kinematics
  (dijelaskan lebih lanjut di algoritma berikutnya)
Else if ID pemrosesan sama dengan 2
  //Mode Forward Kinematics
  Reset logout counter
  Reset kadaluarsa cookies
  Ambil nilai Theta 1 - 5
  If masukan derajat > 0 maka arah ke kanan
    Nilai derajat sama dengan nilai absolut masukan derajat
    Nilai arah sama dengan 255
  If masukan derajat < 0 maka arah ke kiri
    Nilai derajat sama dengan nilai absolut masukan derajat
    Nilai arah sama dengan 0
Else if ID pemrosesan sama dengan "eon"
  //Mode Darurat
  Reset logout counter
  Reset kadaluarsa cookies
  Modifikasi berkas "kdt.txt" untuk memberhentikan sistem
Else if ID pemrosesan sama dengan "reset"
  //Reset posisi robot
  Reset logout counter
  Reset kadaluarsa cookies
  Modifikasi berkas "kdt.txt" untuk memosisikan ke posisi
  default

```

Pemrosesan yang akan dijelaskan pertama kali adalah pemrosesan mode kontrol robot berbasis *inverse kinematics*. Sebagai contoh, disini sistem diberi masukan untuk menggerakkan robot ke koordinat $x = 1$, $y = 2$ dan $z = 3$. Sistem kemudian akan menerjemahkan koordinat tadi untuk kemudian menghitung berapa derajat dan kemana arah tiap-tiap *joint* harus bergerak.

Setelah koordinat didapat dari klik pengguna, maka selanjutnya ketika tombol proses ditekan akan muncul *window* konfirmasi sebagai berikut:



Gambar 3-14. Notifikasi saat akan eksekusi *Inverse Kinematics*

Apabila pengguna sudah yakin dengan masukan yang akan diproses. Maka selanjutnya data masukan akan dieksekusi. Masukan berupa koordinat tadi akan dihitung oleh sistem untuk kemudian diterjemahkan ke dalam bentuk data berupa

derajat dan arah yang akan dikirim ke mikrokontroler pengontrol robot. Hasilnya dapat terlihat seperti pada **Gambar 3-15**.

Namun pada kenyataannya karena pengembangan sistem belum sampai pemrograman *inverse kinematics*, maka algoritma yang dikembangkan untuk memproses data masukan untuk mode ini ialah sebagai berikut.

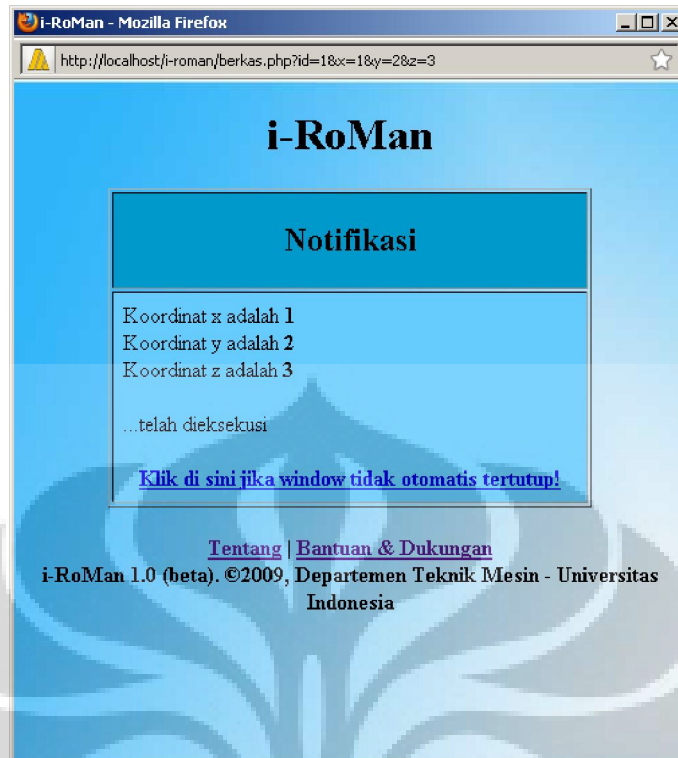
Algoritma Pemrograman

```
Ambil nilai x dan y
//Koordinat X
If nilai x>250
    Joint 1 sama dengan x-250, arahnya 255
    Joint 5 sama dengan x-250, arahnya 255
Else if nilai x<250
    Joint 1 sama dengan 250-x, arahnya 0
    Joint 5 sama dengan 250-x, arahnya 0

//Koordinat Y
If nilai y>80
    Joint 2 sama dengan y-80, arahnya 0
    Joint 3 sama dengan y-80, arahnya 0

    Joint 4 sama dengan y-80, arahnya 0
If nilai y<80
    Joint 2 sama dengan 80-y, arahnya 255
    Joint 3 sama dengan 80-y, arahnya 255
    Joint 4 sama dengan 80-y, arahnya 255
```

Algoritma di atas mencoba untuk memberikan informasi pergerakan untuk kelima *joint* robot dari masukan x, y dan z yang diberikan. Jika dilihat dari algoritmanya, maka nilai yang dibentuk hanyalah agar masukan yang diberikan (koordinat x, y dan z) tidak melebihi jangkauan sistem sehingga dapat diproses. Karena jika tidak, maka sistem akan menolak untuk memprosesnya.



Gambar 3-15. Notifikasi hasil contoh eksekusi *Inverse Kinematics*

- **Mode *Forward Kinematics***

Hampir sama dengan mode berbasis *inverse kinematics*, mode berbasis *forward kinematics* juga dilakukan konfirmasi terlebih dahulu terhadap masukan yang diberikan oleh pengguna. Bedanya karena mode ini menggerakkan robot dengan memberi perintah untuk tiap-tiap *joint*-nya, disini pengguna harus mengisi secara manual berapa derajat dan ke mana arah tiap-tiap *joint* harus bergerak.

Sebagai contoh, disini diperlihatkan data yang diberikan kepada sistem adalah:

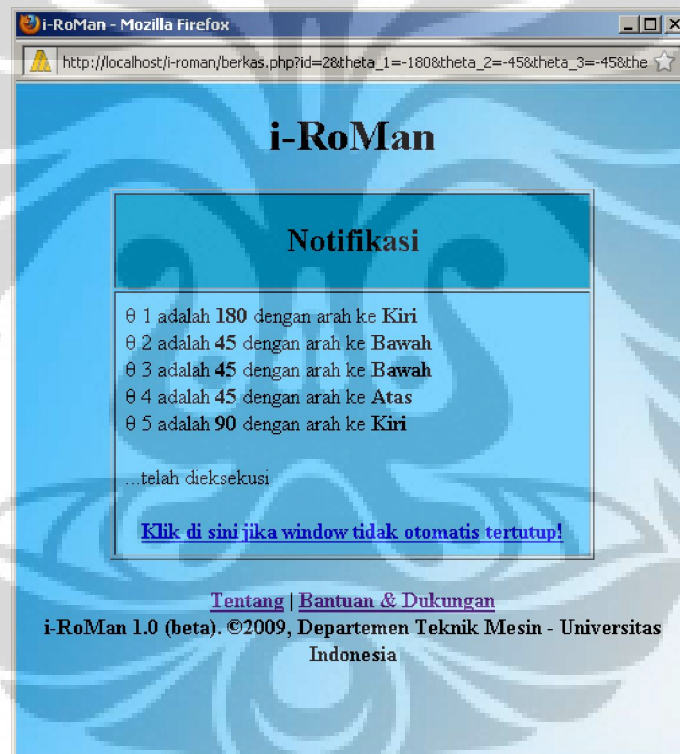
1. Joint 1: -180, berarti 180 derajat ke arah kiri.
2. Joint 2: -45, berarti 45 derajat ke arah bawah.
3. Joint 3: -45, berarti 45 derajat ke arah bawah.
4. Joint 4: 45, berarti 45 derajat ke arah atas.
5. Joint 5: -90, berarti 90 derajat ke arah kiri.

Penjelasan arah gerak dapat dilihat pada **Gambar 3-6**.



Gambar 3-16. Notifikasi saat akan eksekusi *Forward Kinematics*

Jika masukan telah dikonfirmasi oleh pengguna, maka data masukan tersebut akan diproses kemudian dieksekusi oleh sistem. Nilai derajat didapat dengan mengabsolutkan semua nilai. Hasilnya akan ditampilkan pada *window* notifikasi, lengkap dengan arah masing-masing *joint*.



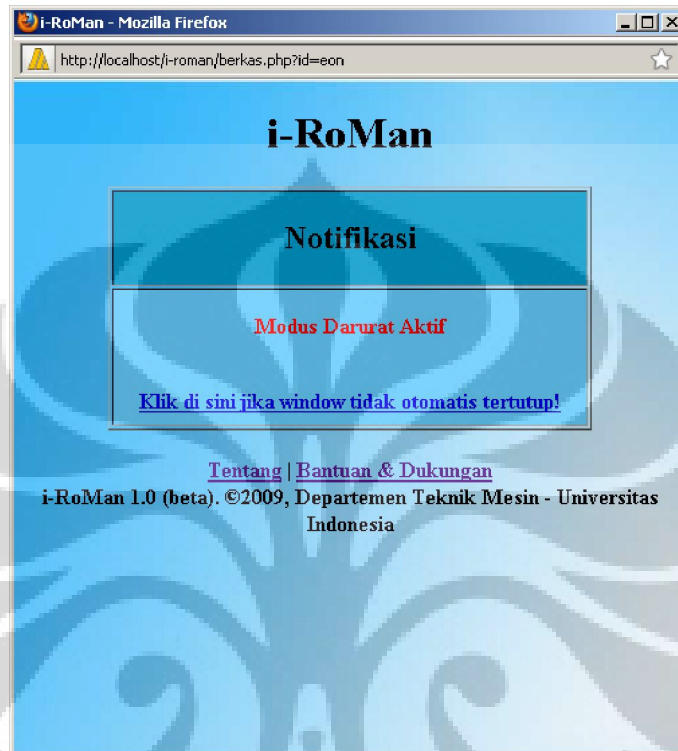
Gambar 3-17. Notifikasi contoh hasil eksekusi *Forward Kinematics*

▪ Mode Darurat dan Reset Posisi

Selain memroses mode *inverse kinematics* dan *forward kinematics*, algoritma pemrosesan masukan juga memroses kondisi darurat dan *reset* posisi.

Kondisi darurat dikontrol penuh eksekusinya oleh pengguna melalui klik tombol darurat ataupun penekanan tombol “Esc” pada keyboard. Ketika mode ini aktif,

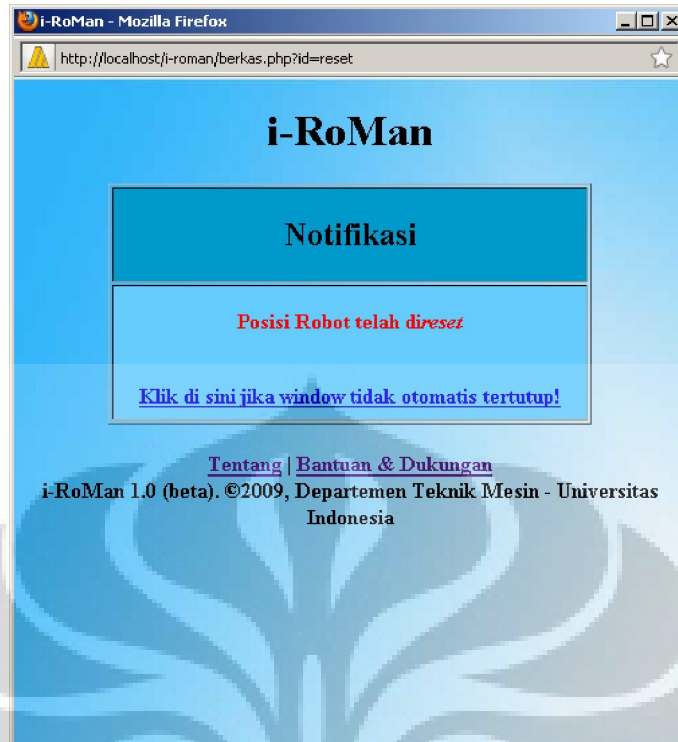
dokumen “berkas.php” akan dipanggil dengan ID proses “eon” untuk menuliskan informasi ke berkas “kdt.txt” yang nantinya akan memberi tahu sistem untuk menghentikan segala proses yang sedang berlangsung.



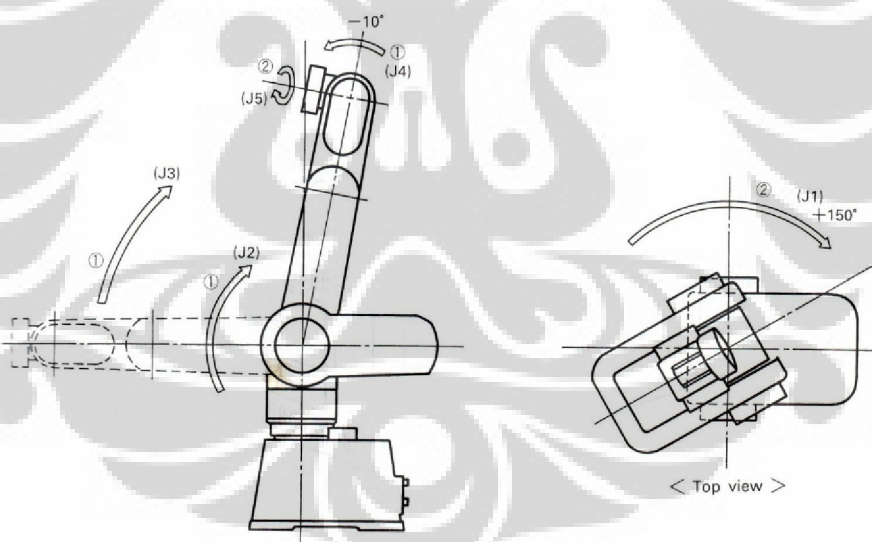
Gambar 3-18. Notifikasi saat pengaktifan kondisi darurat

Untuk mode *reset* posisi, hanya dilakukan melalui klik-kiri pada link “Reset Posisi Robot” di halaman kontrol utama. Sama dengan mode darurat, dokumen “berkas.php” nantinya akan dipanggil dengan ID proses “reset” untuk memberi tahu sistem untuk memposisikan robot ke posisi *default*.

Selain dapat dijalankan secara manual, mode *reset* posisi ini juga dipanggil secara otomatis saat pengguna masuk ke halaman kontrol utama.



Gambar 3-19. Notifikasi *reset* posisi robot



Gambar 3-20. Posisi *default* robot [10]

3.5.2 Struktur Data Keluaran

Untuk berkomunikasi antara *web* dengan mikrokontroler digunakan media yang netral dan paling mudah yaitu berkas teks yang tidak terformat namun terstruktur. Hal ini guna memudahkan pengecekan berkas teks dengan program *viewer* apapun.

Dari segi struktur penulisan data, berkas teks ditulis dengan struktur lima baris dan tiga kolom dengan karakter “spasi” sebagai penanda pergantian kolom. Lima baris merepresentasikan lima motor yang terdapat pada robot Movemaster dan tiga kolom merepresentasikan variabel-variabel yang harus diinformasikan kepada mikrokontroler pengontrol robot yaitu berupa (kolom 1, kolom 2, kolom 3) kecepatan motor, arah motor, dan sudut pergerakan.

Data kecepatan yang diberikan oleh *server* kepada mikrokontroler adalah 180. Kecepatan ini adalah *preset* sistem pada *server* dan dapat dimanipulasi oleh program pada mikrokontroler sesuai kebutuhan dan pergerakan robot.

a ₁ b ₁ c ₁	a	= kecepatan motor
a ₂ b ₂ c ₂	b	= arah pergerakan
a ₃ b ₃ c ₃	c	= derajat pergerakan
a ₄ b ₄ c ₄	1-5	= nomor motor
a ₅ b ₅ c ₅		

(a)

```

D:\WWW\htdocs\i-RoMan\kdt.txt - Notepad++
File Edit Search View Format Language Settings Macro
Run TextFX Plugins Window ?
kdt.txt
1 180 0 180
2 180 0 45
3 180 0 45
4 180 255 45
5 180 0 90
Ln : 1 Col : 1 Sel : UNIX ANSI INS
  
```

(b)

Gambar 3-21. (a) Struktur data pada berkas yang digunakan untuk berkomunikasi antara komputer dengan mikrokontroler. (b) Contoh berkas

Khusus untuk mode darurat dan *reset* posisi robot, sistem tetap menggunakan berkas “kdt.txt” namun nilainya saja yang diubah. Untuk mode darurat, semua variabel untuk tiap-tiap motor memiliki nilai 0 (nol) sedangkan untuk mode *reset* posisi robot, berkas memiliki nilai 1 (satu).

0 0 0	1 1 1
0 0 0	1 1 1
0 0 0	1 1 1
0 0 0	1 1 1
0 0 0	1 1 1

(a) (b)

Gambar 3-22. Isi berkas “kdt.txt” untuk (a) Mode darurat. (b) Mode reset

3.6 Implementasi Sistem

Sistem yang telah dibangun tentunya harus diimplementasikan disuatu infrastruktur agar dapat diakses oleh pengguna. Oleh karena itu, sistem ini diimplementasikan ke dalam sebuah komputer yang dijadikan sebagai *server* yang terletak di Laboratorium Manufaktur dan Otomasi DTM UI.

3.6.1 Instalasi

▪ Piranti Lunak

1. *Install web server* ke komputer *server*. Setelah *terinstall*, *web server* akan otomatis aktif dan berjalan saat *server* melakukan *startup* Microsoft WindowsXP.
2. *Install interpreter PHP* ke komputer *server*. Disarankan *diinstall* ke dalam *folder* yang sama dengan *folder* yang sama dengan *web server*. Selanjutnya menghubungkan antara *interpreter PHP* ini dengan *web server* sehingga setiap dokumen PHP yang dipanggil dan dieksekusi, *web server* akan memanggil *interpreter PHP* untuk memrosesnya.
3. *Install web camera server* ke komputer *server* sebanyak dua kali ke *path* (tempat) yang berbeda. Sehingga terdapat dua program yang sama di dalam *server*. Masing-masing *web camera server* mengontrol satu *web camera* dan memberikan satu alamat *port* khusus untuk pengaksesan *web camera*.

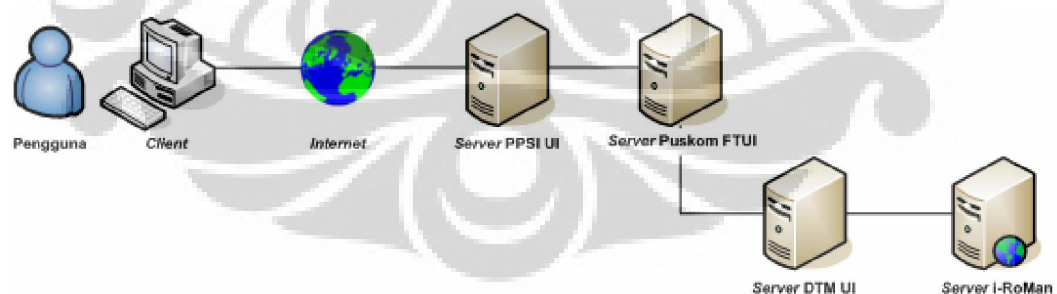
▪ Piranti Keras

Dua buah *web camera* dikoneksikan dengan komputer yang telah disiapkan untuk menjadi *server* melalui *port* USB. Satu *web camera* diletakkan tepat di atas robot (x-z) dan satu lagi berada di sisi samping robot (x-y). Setelah itu, komputer *server* dihubungkan ke jaringan melalui sebuah kabel UTP-8 ke sebuah *router* jaringan yang terkoneksi ke *intranet* DTM UI.

3.6.2 Metode Akses

Agar dapat diakses melalui *internet*, diperlukan IP statis yang didedikasikan khusus untuk *server* tempat i-RoMan diinstall. IP tersebut harus diset supaya permintaan koneksi dapat langsung diteruskan oleh *server* PPSI UI ke *server* i-RoMan. Karena jika *server* PPSI UI hanya dijadikan sebagai *proxy server*, maka semua koneksi yang datang ke sistem akan dianggap berasal dari satu alamat IP. Hal ini dapat mengacaukan manajemen sesi yang telah dibuat untuk memproteksi sistem.

Situs <http://mech.eng.ui.ac.id> yang telah terinstall sebelumnya di Puskom Fakultas Teknik UI (Puskom FTUI) dibuatkan *virtual path* bernama “i-RoMan”. Oleh karena itu, yang meneruskan permintaan koneksi dari *internet* sebenarnya adalah *server* Puskom FTUI baru kemudian ke *server* i-RoMan. Sehingga sistem dapat langsung diakses melalui *internet* dengan alamat <http://mech.eng.ui.ac.id/i-RoMan>.



Gambar 3-23. Jalur lalu lintas data dari dan ke sistem i-RoMan

Namun hingga laporan skripsi ini dibuat, *server* masih belum diizinkan untuk diakses langsung dari luar intranet DTM UI.

Bab 4. Pengujian dan Analisa Sistem

Bab ini menjelaskan tentang pengujian dan analisa dari sistem komunikasi dan GUI yang telah dibangun termasuk perangkat yang digunakan untuk uji coba beserta *setup* dari masing-masing perangkat tersebut. Semua fungsi-fungsi yang telah diprogram diuji coba kemudian hasil yang didapat termasuk anomali sistem yang mungkin terjadi dianalisa.

4.1 Perangkat Uji Coba

Untuk menguji coba sistem digunakan beberapa komputer yang bertindak sebagai *client* untuk kemudian mengakses *server* dan menjalankan beberapa fungsi yang telah dijelaskan pada bab-bab sebelumnya.

Sistem ini sendiri telah dirancang langsung dapat berjalan pada komputer pengguna bahkan pada komputer yang baru sekalipun dengan piranti lunak bawaan pabrik. Sehingga tidak dibutuhkan tambahan apapun. Spesifikasi komputer *client* yang digunakan untuk pengujian adalah sebagai berikut:

Tabel 4-1. Spesifikasi komputer *desktop client*

Jenis	Nama
Prosesor	Intel Pentium D Dual-core 3 GHz
RAM	1 GB DDR2-SDRAM
HDD	80 GB
Sistem Operasi	Microsoft Windows XP SP 2
<i>Web Browser</i>	Mozilla Firefox dan Internet Explorer
Koneksi	100Mbps (kabel)

Tabel 4-2. Spesifikasi komputer *laptop client 1*

Jenis	Nama
Merk/Tipe	HP Compaq/Presario V2406AU
Prosesor	AMD Sempron 3000+
RAM	768 MB DDR-SDRAM

HDD	40 GB
Sistem Operasi	Microsoft Windows XP SP 2
<i>Web Browser</i>	Mozilla Firefox dan Internet Explorer
Koneksi	54Mbps (nirkabel)

4.2 Kompatibilitas Web Browser

Proses uji coba sistem dilakukan melalui perangkat yang berbeda-beda. Begitu pula dengan *web browser* yang digunakan. Banyaknya *web browser* yang tersedia di pasaran mengharuskan sistem ini mampu dijalankan di semua *web browser* yang ada, sehingga menghilangkan ketergantungan penggunaan sistem pada *web browser* tertentu. Untuk melihat kompatibilitas program (terutama JavaScript) yang telah dibuat, maka digunakanlah dua jenis *web browser* yaitu Mozilla Firefox dan Internet Explorer, hal ini dikarenakan pangsa pasar kedua *web browser* di dunia >80% [16].

Tabel 4-3. Kompatibilitas sistem dengan *web browser*

Proses/Bagian	Web Browser	
	Internet Explorer 6.0-8.0	Mozilla Firefox 3.5
Tampilan HTML	√	√
Penangkapan Gambar Web Camera	√	√
Validasi Data Masukan	√	√
Manajemen Sesi*	√	√
Penanganan Kesalahan	√	√

keterangan:

√ = dapat dilaksanakan tanpa error

* = dengan catatan

Secara keseluruhan, sistem dapat berjalan dengan baik pada kedua jenis *web browser*. Hanya saja karena terdapat perbedaan *engine* antara keduanya, tampilan

yang muncul di layar komputer pengguna terdapat sedikit perbedaan yang sebenarnya tidak terlalu mencolok yang antara lain seperti *margin* dan jarak spasi antar isi dalam dokumen HTML. Namun dari sisi kinerja sistem, kedua jenis *web browser* dapat menjalankan semua fungsi sistem yang telah dirancang dengan baik.

Terdapat catatan kecil saat pengujian yaitu ketika halaman kontrol dimuat ulang (*refresh/reload*) atau komputer pengguna yang sedang aktif crash, maka manajemen sesi akan menjadi rusak. Di satu sisi, pengguna tersebut tetap dapat membuka halaman kontrol, namun IP pengguna yang tercatat di *server* dan cookies yang tersimpan di komputer pengguna akan terhapus. Hal ini dikarenakan tidak adanya *handler* khusus untuk mendeteksi *page refresh/reload* pada halaman kontrol utama, sehingga aksi tersebut akan dianggap sama dengan aksi penutupan *window*/pengaksesan halaman *web* lain ketika halaman kontrol utama masih aktif.

Akibatnya, pengguna yang lain tidak dapat mengakses sistem walaupun nantinya pengguna yang mengakibatkan rusaknya manajemen sesi tersebut sudah tidak aktif lagi.

4.3 Streaming Gambar Web Camera

Streaming gambar dari *web camera* dapat berjalan dengan baik. Hal ini wajar dikarenakan pada komputer *client* yang digunakan untuk uji coba terkoneksi ke *intranet* DTM UI melalui perangkat *router* jaringan yang sama dengan komputer *server* yang memiliki kecepatan koneksi teoritis sebesar 100Mbps atau sekitar 12,8 MBps, jauh melebihi batas minimum kecepatan penggunaan sistem yang “hanya” sebesar 168 KBps.

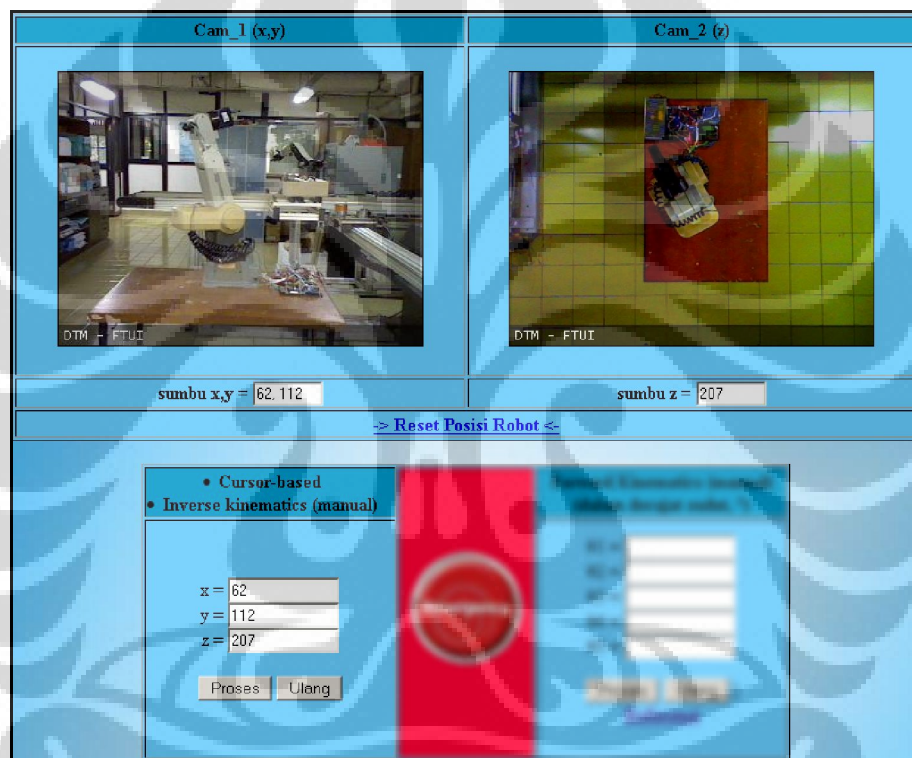
Pengujian juga dilakukan melalui sebuah *laptop* yang memakai koneksi *wireless* ke *intranet* DTM UI (berbeda *router*, namun masih dalam satu jaringan *intranet* DTM UI) dan hal yang sama juga terjadi yaitu pemuatan *streaming* gambar dari *web camera* dapat berjalan lancar.

4.4 Inverse Kinematics

Sejauh ini sistem belum dapat mengaplikasikan mode berbasis *inverse kinematics* secara sempurna. Hal ini dikarenakan belum adanya *engine* untuk mengubah data koordinat yang ditentukan baik melalui masukan langsung

ataupun berdasarkan klik *cursor mouse (cursor-based)* atau dengan kata lain mentrasformasikan dari *web-windows coordinate* menjadi *robot actual cartesian domain*.

Oleh karena itu, untuk keperluan pengujian sistem hanya digunakan semacam *dummy data* sehingga reaksi sistem ketika mode berbasis *inverse kinematics* dijalankan dapat dilihat. *Dummy data* ini dihasilkan dari pemrograman yang telah ditanam sebelumnya ke bagian pemrosesan. Contoh nilai masukan yang dibuat *dummy datanya* dapat dilihat pada **Gambar 4-1**.



Gambar 4-1. Uji coba mode *Inverse Kinematics*

Ketika dimasukkan data berupa koordinat $x = 62$, $y = 112$ dan $z = 207$ ke form isian *inverse kinematics* di bagian sebelah kiri halaman kontrol utama kemudian diproses, maka *dummy data* yang keluar dan tertulis didalam berkas kontrol adalah:

180 0 188
180 255 32
180 255 32
180 255 32
180 0 188

Gambar 4-2. *Dummy data* berupa $x = 62$, $y = 112$ dan $z = 207$

Data di atas memerintahkan:

1. Joint 1 bergerak ke arah kiri sebesar 188 derajat.
2. Joint 2 bergerak ke arah atas sebesar 32 derajat.
3. Joint 3 bergerak ke arah atas sebesar 32 derajat.
4. Joint 4 bergerak ke arah atas sebesar 32 derajat.
5. Joint 5 bergerak ke arah kiri sebesar 188 derajat.

Hasil dari percobaan menunjukkan bahwa sistem dapat memberikan tanggapan dan menggerakkan robot sesuai data masukan di atas.

4.5 Forward Kinematics

Mode berbasis *forward kinematics* telah dapat dilakukan dengan baik oleh sistem dikarenakan hasil yang dikeluarkan dari hasil pemrosesan sudah sesuai dengan algoritma yang dibuat. Sebagai contoh sistem diberi masukan (posisi awal robot dari posisi *default*, yaitu ketika semua *joint* menyentuh *limit switch*nya seperti pada **Gambar 3-20**) untuk *joint* 1-3 = -90 dan *joint* 4-5 = 90.



Gambar 4-3. Uji Coba mode *Forward Kinematics*

Maka data yang tertulis di berkas kontrol “kdt.txt” setelah masukan tadi dieksekusi dapat dilihat pada Gambar 4-4.

```
180 0 90
180 0 90
180 0 90
180 255 90
180 255 90
```

Gambar 4-4. Isi berkas kontrol untuk pergerakan *Joint 1-3 = -90* dan *Joint 4-5 = 90*

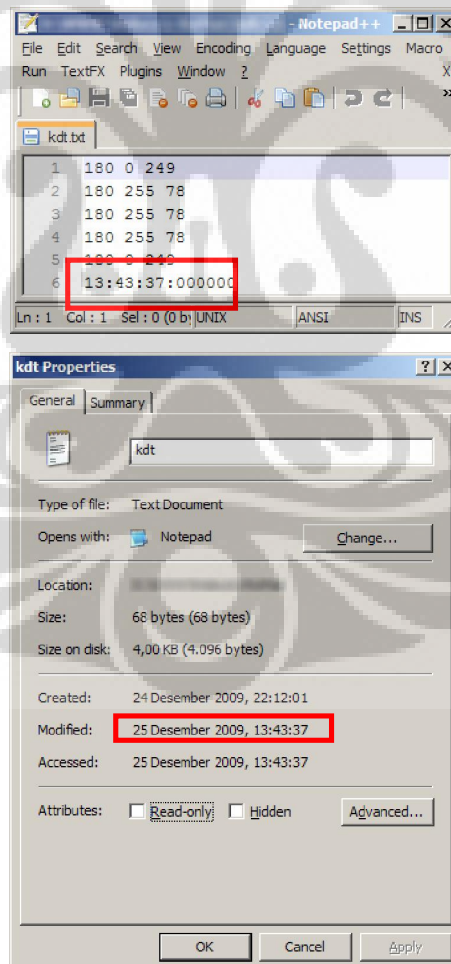
Setelah dieksekusi, sistem mampu memberikan tanggapan dan memroses masukan yang diberikan dengan menggerakkan robot sesuai dengan data di atas, terutama untuk arah-arah pergerakan masing-masing *joint*.

4.6 Unjuk Kerja

Pengujian yang terakhir dilakukan ialah pengujian unjuk kerja sistem. Faktor yang diuji ialah seberapa cepat (dalam detik) tiap masukan yang diberikan diproses. Metode yang digunakan ialah dengan mencatatkan waktu ketika pemrosesan dimulai ke dalam berkas “kdt.txt” kemudian membandingkannya dengan waktu ketika berkas tersebut dibuat.

Tabel 4-4. Waktu pemrosesan masukan hingga menjadi berkas kontrol

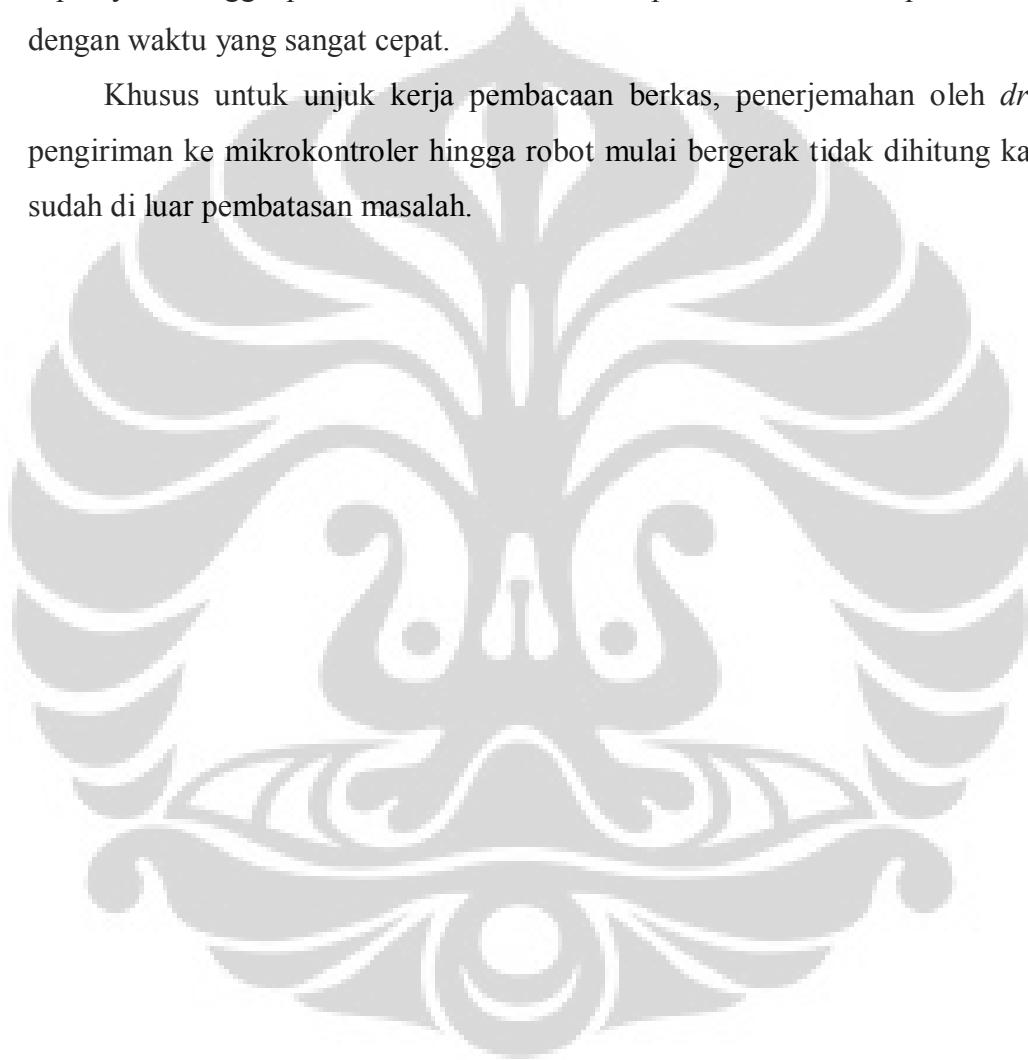
Konektivitas	Waktu Pemrosesan (detik)					
	I	II	III	IV	V	VI
<i>Local system</i> (http://localhost)	0	0	0	0	0	0
Melalui LAN/WLAN	0	0	0	0	0	0



Gambar 5. Metode penghitungan waktu pemrosesan

Dapat dilihat bahwa waktu yang dibutuhkan untuk memproses masukan hingga menjadi berkas sangatlah cepat, hanya 0 (nol) detik. Hal ini dikarenakan masukan yang diproses sangatlah sedikit (paling banyak lima nilai di mode berbasis *forward kinematics*). Selain itu algoritma pemrosesan yang digunakan relatif tidak kompleks. Terlebih lagi kecepatan prosesor komputer saat ini sudah sedemikian cepatnya sehingga pemrosesan data masukan pada sistem ini dapat dilakukan dengan waktu yang sangat cepat.

Khusus untuk unjuk kerja pembacaan berkas, penerjemahan oleh *driver*, pengiriman ke mikrokontroler hingga robot mulai bergerak tidak dihitung karena sudah di luar pembatasan masalah.



Bab 5. Kesimpulan dan Saran Penelitian Lebih Lanjut

5.1 Kesimpulan

Sistem mampu dan berhasil untuk mengontrol robot secara *realtime* dan *termonitor* dengan bantuan gambar yang *distreaming* dari dua buah *web camera* yang digunakan melalui *web interface* yang diakses oleh pengguna. *Interface* sistem ini sendiri dibangun dengan satu kumpulan aplikasi yang berbasiskan *web* dengan tiga bahasa *web scripting* yaitu HTML, JavaScript dan PHP. Untuk mode berbasis *forward kinematics*, sistem sudah dapat melakukannya dengan baik. Namun sebaliknya, untuk mode berbasis *inverse kinematics* sistem belum mampu melakukannya dengan baik.

Sistem tersebut mampu berkomunikasi dengan mikrokontroler pengontrol robot dengan perantara kabel USB yang terhubung ke *server* melalui berkas teks berisi perintah-perintah pergerakan robot. Berkas teks tersebut diterjemahkan terlebih dahulu oleh *driver* robot yang terdapat pada *server*. Kemudian *driver tersebut* akan mengirimkannya ke mikrokontroler untuk dieksekusi lebih lanjut dalam bentuk pergerakan robot.

5.2 Saran Penelitian Lebih Lanjut

Dari hasil penelitian dapat disimpulkan beberapa saran untuk pengembangan sistem selanjutnya, yaitu:

- Penggunaan *IP camera*. Penggunaan *IP camera* dapat mengurangi beban dari *server* karena *web camera server* sudah tertanam di dalam *camera* tersebut.
- Penggunaan jaringan *internet* yang cepat. Sistem *i-RoMan* termasuk sistem yang haus *bandwidth*, terutama untuk *streaming* gambar dari *web camera*. Untuk itu tersedianya jalur data yang lebar dan cepat memungkinkan pengurangan kompensasi-kompensasi selama pengembangan sistem diterapkan, sehingga pengalaman penggunaan pun menjadi lebih cepat dan nyaman.

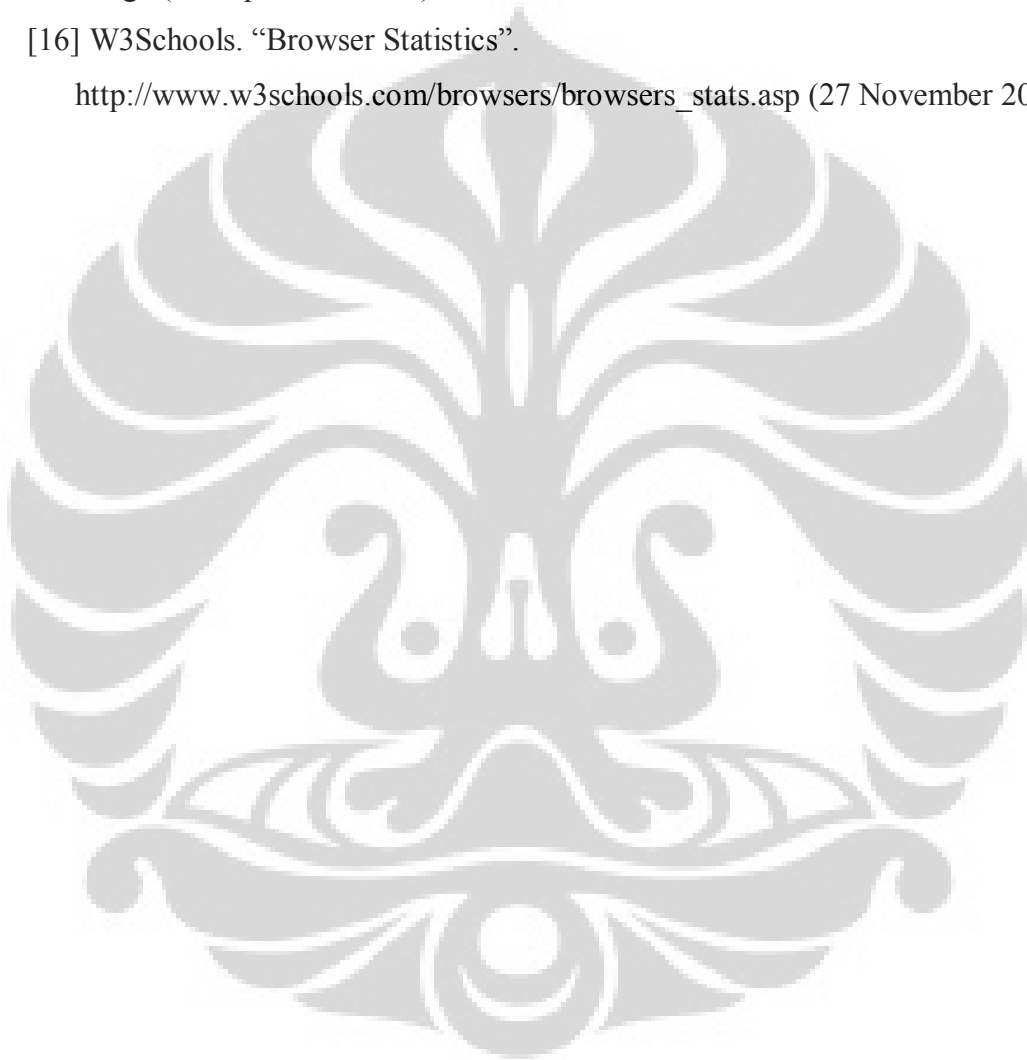
- Pembuatan *account admin* khusus untuk mengantisipasi *error* yang terjadi dalam sistem. Misalnya saja pada kasus rusaknya manajemen sesi yang telah dijelaskan sebelumnya.
- Menyatukan komputer dengan *control unit*. Jika komputer yang digunakan sebagai *server* dapat dibuat berukuran ringkas. Maka infrastruktur yang dibutuhkan menjadi lebih sederhana. Terlebih jika koneksi yang digunakan ialah WLAN, maka robot dapat dibuat menjadi *mobile*.



Daftar Referensi

- [1] APJII. “Statistik APJII Updated Desember 2007”.
<http://www.apjii.or.id/dokumentasi/statistik.php?lang=ind> (15 Nov 2009).
- [2] Fielding, et al. “Hypertext Transfer Protocol -- HTTP/1.1”.
<http://tools.ietf.org/html/rfc2616> (12 November 2009)
- [3] Howstuffworks. "How Web Servers Work".
<http://www.howstuffworks.com/web-server.htm> (12 November 2009)
- [4] Answers.com. “HTML: Definition from Answers.com”.
<http://www.answers.com/topic/html> (12 November 2009)
- [5] Alamsyah, Andry. 2003. Pengantar JavaScript: IlmuKomputer.com
- [6] SoftwareProjects. “What Is PHP?”. <http://www.softwareprojects.org/php-what-is-01.htm> (12 November 2009)
- [7] O’Reilly Media. “An Introduction to PHP”.
http://www.oreillynet.com/pub/a/php/2001/02/22/php_foundations.html (12 November 2009)
- [8] Purbo, Onno W. “Pengertian Server”.
<http://cybertech.cbn.net.id/cbprtl/cybertech/detail.aspx?x=Tech+Talk&y=cybertech|0|3|11> (11 Desember 2009)
- [9] International Standard Organization. 1994. ISO 8373-1994 - Manipulating industrial robots.
- [10] Mitsubishi Electric Corp. Industrial Micro-Robot System RV-M1 Instruction Manual.
- [11] The MathWorks Inc. “Represent assembled revolute joint with one rotational degree of freedom”.
<http://www.mathworks.com/access/helpdesk/help/toolbox/phymod/mech/ref/revolute.html> (27 November 2009)
- [12] The MathWorks Inc. “Represent prismatic joint with one translational degree of freedom”.
<http://www.mathworks.com/access/helpdesk/help/toolbox/phymod/mech/ref/prismatic.html> (27 November 2009)

- [13] TikZ Example. “Annotated manipulator”.
<http://www.texample.net/tikz/examples/three-link-annotated> (24 November 2009)
- [14] <http://www.rixan.com/Portals/0/RV-M1.jpg> (27 November 2009)
- [15] BYTES. “x,y coordinates from an `<input type='image'>`”.
<http://bytes.com/topic/javascript/answers/527139-x-y-coordinates-input-type-image> (24 September 2009)
- [16] W3Schools. “Browser Statistics”.
http://www.w3schools.com/browsers/browsers_stats.asp (27 November 2009)



Lampiran

- Dokumen-dokumen PHP
- Dokumen-dokumen JavaScript



index.php

```

<?php
/*-----
Copyright 2009. Anom Tejo Pratomo
Departemen Teknik Mesin - Universitas Indonesia
-----*/
error_reporting(0);

/* Kontrol Pengguna */
require('kontrol_pengguna/sas.php'); // Manajemen sesi pengguna
require('kontrol_pengguna/cek_kecepatan.php'); // Pendeteksi kecepatan akses internet
pengguna
if ($kecepatan < 360) {
    limiter();
    exit;
}
?>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
    <title>i-RoMan</title>
    <meta name="author" content="Departemen Teknik Mesin - Universitas
Indonesia">
    <meta name="editor" content="Notepad++">
    <meta name="description" content="Web-based Robot Manipulation">
    <meta name="keywords" content="web-based, robot, manipulation">
    <meta http-equiv="cache-control" content="no-cache">
    <meta http-equiv="pragma" content="no-cache">
    <link rel="shortcut icon" href="images/favicon.ico">
    <script type="text/javascript" src="js/inti.js"></script>
    <script type="text/javascript" src="js/kontrol_pengguna.js"></script>
    <script type="text/javascript" src="js/webcam.js"></script>
</head>
<body style="background-image:url('images/background.jpg');background-attachment:
fixed" onload="reload1(4);reload2(4);reset()" onkeypress="keycheck(event);">
<h1 align="center">i-RoMan</h1>
<table width="84%" align="center" border="1" bgcolor="#66ccff">
    <tr>
        <th bgcolor="#0099cc">Cam_1 (x,y)</th><th bgcolor="#0099cc">Cam_2
(z)</th>
    </tr><tr>
        <td width="50%" align="center"><br><br><br></td>
        <td width="50%" align="center"><br><br><br></td>
    </tr><tr>
        <form>
            <td align="center"><b>sumbu x,y = </b><input type="text" id="cam_1"
size="6"><br></td>
            <td align="center"><b>sumbu z = </b><input type="text" id="cam_2"
size="6"><br></td>
        </tr>

```

```

        </form>
    </tr>
    <tr>
        <td align="center" colspan="2"><a href="#" onclick="reset();"><b>->
Reset Posisi Robot <-</b></a></td>
    </tr>
</table>
<br>
<table width="60%" align="center" border="1" bgcolor="#66ccff">
    <th bgcolor="#0099cc"><li>Cursor-based<br><li>Inverse kinematics
(manual)</th>
    <th bgcolor="#ff0033" rowspan="2"><input type="image"
src="images/emrgncy.gif" onclick="darurat()"></th>
    <th bgcolor="#0099cc">Forward Kinematics (manual)<br>(dalam derajat sudut,
&deg;)</th>
    <tr>
        <td align="center" rowspan="2" style="padding: 12px" width="50%">
            <form>
                x =
                <input type="text" id="cam_1_x_diklik" name="x" onblur="validasi(this)"
size="12"><br>
                y =
                <input type="text" id="cam_1_y_diklik" name="y" onblur="validasi(this)"
size="12"><br>
                z =
                <input type="text" id="cam_2_z_diklik" name="z" onblur="validasi(this)"
size="12"><br>
                <br><input type="button" value="Proses" onClick="konfirmasi(1)">
                <input type="reset" value="Ulang"><br>
            </form>
        </td>
        <td align="center" style="padding: 12px" width="50%">
            <form>
                &theta;1 =
                <input type="text" id="theta_1" name="theta_1" onblur="validasi(this)"
size="12"><br>
                &theta;2 =
                <input type="text" id="theta_2" name="theta_2" onblur="validasi(this)"
size="12"><br>
                &theta;3 =
                <input type="text" id="theta_3" name="theta_3" onblur="validasi(this)"
size="12"><br>
                &theta;4 =
                <input type="text" id="theta_4" name="theta_4" onblur="validasi(this)"
size="12"><br>
                &theta;5 =
                <input type="text" id="theta_5" name="theta_5" onblur="validasi(this)"
size="12"><br>
                <p align="center"><input type="button" value="Proses"
onClick="konfirmasi(2)">
                <input type="reset" value="Ulang"> | <a href="images/sketsa.jpg"
target="_blank"><b>Referensi</b></a></p>
            </td>
        </tr>
    </tr>
</table>

```



```
<br>
<?php include("templates/footer_logged.php"); ?>
```

berkas.php

```
<?php
/*-----
Copyright 2009. Anom Tejo Pratomo & Adhitya Bayu W.
Departemen Teknik Mesin - Universitas Indonesia
-----*/
error_reporting(0);

setcookie('sascookie',$_COOKIE['sascookie'],time()+540);
require('kontrol_pengguna/sas.php'); // Manajemen sesi pengguna
include("templates/header.php");

// Jika berkas ini dipanggil secara langsung
if (isset($_REQUEST[id])) {
    // Penanganan Kondisi Darurat
    if ($_REQUEST[id] == "eon") {
        $fh = fopen('kdt.txt', 'w') or die ("File tidak dapat dibuka!");
        fwrite($fh, "0 0 0\n0 0 0\n0 0 0\n0 0 0\n0 0 0");
        fclose($fh);
        ?>
        <p align="center"><blink><font color="red"><b>Modus Darurat
Aktif</b></font></blink></p>
        <script>setTimeout('window.close()',4000);</script>
        <center><br><a href="javascript:window.close();"><b>Klik di sini jika
window tidak otomatis tertutup!</b></a></center>
        <?php
    }

    // Reset Posisi Robot
    else if ($_REQUEST[id] == "reset") {
        $fh = fopen('kdt.txt', 'w') or die ("File tidak dapat dibuka!");
        fwrite($fh, "1 1 1\n1 1 1\n1 1 1\n1 1 1\n1 1 1");
        fclose($fh);
        ?>
        <p align="center"><blink><font color="red"><b>Posisi Robot telah
di<i>reset</i></b></font></blink></p>
        <script>setTimeout('window.close()',4000);</script>
        <center><br><a href="javascript:window.close();"><b>Klik di sini jika
window tidak otomatis tertutup!</b></a></center>
        <?php
    }

    // Penerjemahan dan Pemrosesan Data Masukan
    else {
        if ($_REQUEST[theta_1] > 0) {
            $arah1 = "Kanan";$nulisarah1 = 255;
        } else {
            $arah1 = "Kiri";$nulisarah1 = 0;
        }
        if ($_REQUEST[theta_2] > 0) {
```

```

    $arah2 = "Atas";$nulisarah2 = 255;
} else {
    $arah2 = "Bawah";$nulisarah2 = 0;
}
if ($_REQUEST[theta_3] > 0) {
    $arah3 = "Atas";$nulisarah3 = 255;
} else {
    $arah3 = "Bawah";$nulisarah3 = 0;
}
if ($_REQUEST[theta_4] > 0) {
    $arah4 = "Atas";$nulisarah4 = 255;
} else {
    $arah4 = "Bawah";$nulisarah4 = 0;
}
if ($_REQUEST[theta_5] > 0) {
    $arah5 = "Kanan";$nulisarah5 = 255;
} else {
    $arah5 = "Kiri";$nulisarah5 = 0;
}
}
$nulissudut1 = abs($_REQUEST[theta_1]);
$nulissudut2 = abs($_REQUEST[theta_2]);
$nulissudut3 = abs($_REQUEST[theta_3]);
$nulissudut4 = abs($_REQUEST[theta_4]);
$nulissudut5 = abs($_REQUEST[theta_5]);

$fh = fopen('kdt.txt', 'w') or die ("File tidak dapat dibuka!");
if ($_REQUEST[id] == 1) {
    $x = $_REQUEST[x];
    $y = $_REQUEST[y];
    $z = $_REQUEST[z];

    if($x > 250){
        $theta_1 = $x - 250;$arah_1 = 255;
        $theta_5 = $x - 250;$arah_5 = 255;
    } else{
        $theta_1 = 250 - $x;$arah_1 = 0;
        $theta_5 = 250 - $x;$arah_5 = 0;
    }
    if($y > 250){
        $theta_2 = abs($y - 80);$arah_2 = 0;
        $theta_3 = abs($y - 80);$arah_3 = 0;
        $theta_4 = abs($y - 80);$arah_4 = 0;
    } else {
        $theta_2 = abs(80 - $y);$arah_2 = 255;
        $theta_3 = abs(80 - $y);$arah_3 = 255;
        $theta_4 = abs(80 - $y);$arah_4 = 255;
    }
    fwrite($fh, "180 $arah_1 $theta_1\n180 $arah_2 $theta_2\n180
    $arah_3 $theta_3\n180 $arah_4 $theta_4\n180 $arah_5 $theta_5");
    echo 'Koordinat x adalah <b>'. $x. '</b><br>';
    echo 'Koordinat y adalah <b>'. $y. '</b><br>';
    echo 'Koordinat z adalah <b>'. $z. '</b><br>';
} else if ($_REQUEST[id] == 2) {
    fwrite($fh, "180 $nulisarah1 $nulissudut1\n180 $nulisarah2
    $nulissudut2\n180 $nulisarah3 $nulissudut3\n180 $nulisarah4 $nulissudut4\n180
    $nulisarah5 $nulissudut5");
}

```

```

        echo '&theta; 1 adalah <b>'. $nulissudut1. '</b> dengan arah ke
<b>'. $arah1. '</b><br>';
        echo '&theta; 2 adalah <b>'. $nulissudut2. '</b> dengan arah ke
<b>'. $arah2. '</b><br>';
        echo '&theta; 3 adalah <b>'. $nulissudut3. '</b> dengan arah ke
<b>'. $arah3. '</b><br>';
        echo '&theta; 4 adalah <b>'. $nulissudut4. '</b> dengan arah ke
<b>'. $arah4. '</b><br>';
        echo '&theta; 5 adalah <b>'. $nulissudut5. '</b> dengan arah ke
<b>'. $arah5. '</b><br>';
    }
    fclose($fh);
    echo '<br>...telah dieksekusi!';
    echo '<script>setTimeout(\'window.close()\',4000);</script>';
    echo '<center><br><a href="javascript:window.close();"><b>Klik di sini jika
window tidak otomatis tertutup!</b></a></center>';
}
} else {
echo '<br>Maaf, Anda tidak dapat mengakses berkas ini secara langsung!<br>';
}
?>
</div></td>
</tr></table>
<?php include("templates/footer_unlogged.php"); ?>
sas.php
<?php
/*-----
Copyright 2005. Zann-Marketing (http://www.zann-marketing.com). Dimodifikasi oleh Anom Tejo Pratomo
Departemen Teknik Mesin - Universitas Indonesia
-----*/

error_reporting(0);

/* Config Section */
$pass      = 'mesin';           // Set the password.
$cookieName = 'sascookie';     // Optional change: Give the cookie a name.
Default is sascookie
$expirytime = time()+540;      // Optional change: Set an expiry time for the
password (in seconds).
$msg       = 'Password salah!'; // Optional change: Error message displayed when
password is incorrect. Default is "Password incorrect".
/* End Config */

/* Logout Stuff - Sept 5, 2005 */
if (isset($_REQUEST['logout'])) {
    $uh = fopen('kontrol_pengguna/kontrol.txt', 'r') or die ("Berkas tidak dapat
dibuka!");
    $ip_tercatat = fread($uh,15);
    fclose($uh);
    if ($ip_tercatat == $_ENV[REMOTE_ADDR]) {
        setcookie($cookieName, "", time()-3600);
    }
    // remove cookie/password

```

```

        $url=str_replace('?logout',',$_SERVER['REQUEST_URI']); //
remove the string "?logout" from the URL
        header('Location: '.$url);
// redirect the browser to original URL
        $uh = fopen('kontrol_pengguna/kontrol.txt', 'w') or die ("Berkas tidak dapat
dibuka!");
        fwrite($uh, 0);
        fclose($uh);
        show_login_page("");
    } else {
        include("templates/header.php"); ?>
        <p align="justify">Maaf, Anda tidak berhak melakukan eksekusi ini!</p>
        </div></td>
        </tr></table>
        <?php include("templates/footer_unlogged.php");
    }
}
/* End Logout Stuff */

/* FUNCTIONS */
$encrypt_pass=md5($pass); // encrypt password

function setmycookie() {
global $cookiename,$encrypt_pass,$expirytime;
    setcookie($cookiename,$encrypt_pass,$expirytime);
}

function show_login_page($msg) {
    $uh = fopen('kontrol_pengguna/kontrol.txt', 'r') or die ("Berkas tidak dapat
dibuka!");
    $ip_tercatat = fread($uh,15);
    fclose($uh);
    include("templates/header.php");
    if ($ip_tercatat != 0) {
        if ($ip_tercatat != $_ENV[REMOTE_ADDR]) {
            echo '<p align="justify">Terdapat sesi pengguna yang sedang aktif, silakan
menunggu beberapa saat lagi.</p>';
        } else if ($ip_tercatat == $_ENV[REMOTE_ADDR] &&
isset($_COOKIE[$cookiename])) {
            //
        } else {
            echo '<p align="justify">Halaman ini diakses dari web browser lain pada
komputer yang sama.</p>';
        }
        echo '</div></td>
        </tr></table>
        <br>';
        include("templates/footer_unlogged.php");
    } else if ($ip_tercatat == 0) { ?>
        <p align="justify">Untuk mengakses halaman kontrol, Anda harus terlebih
dahulu memasukkan password pada form dibawah ini. Harap diingat bahwa setiap login
session berlangsung apabila selama 9 menit (540 detik) tidak ada aktifitas, maka Anda
akan ter-logout secara otomatis.
        </p><hr>
        <center><form action="" method="POST">

```

```

        Password:&nbsp;<input type="password" name="password"
size="20">&nbsp;  
        <input type="submit" value="Login">
        <input type="hidden" name="sub" value="sub">
    </form></center><br>
    <?php
    if (isset($msg)) {
    echo '<center><blink><font
color=red><b>'. $msg.'</b></font></blink></center>';
    }
    ?>
    </div></td>
</tr></table>
<? include("templates/footer_unlogged.php");
}
exit();
}
/* END FUNCTIONS */

/* Cookie Handler */
$errormsg="";
if (substr($_SERVER['REQUEST_URI'],-7)!='sas.php') { // if someone tries to request
sas.php
    if (isset($_POST['sub'])) { // if form has been
submitted
        $submitted_pass=md5($_POST['password']); // encrypt submitted
password
        if ($submitted_pass<>$encrypt_pass) { // if password is incorrect
            $errmsg=$msg;
            show_login_page($errmsg);
        } else { // if password is
correct, create cookie and file control
            setmycookie();
            $uh = fopen('kontrol_pengguna/kontrol.txt', 'w') or die ("Berkas tidak
dapat dibuka!");
            fwrite($uh, $_ENV[REMOTE_ADDR]);
            fclose($uh);
        }
    } else {
        if (isset($_COOKIE[$cookiename])) { // if cookie is set
            if ($_COOKIE[$cookiename]==$encrypt_pass) { // if cookie is correct
                $uh = fopen('kontrol_pengguna/kontrol.txt', 'w') or die ("Berkas
tidak dapat dibuka!");
                fwrite($uh, $_ENV[REMOTE_ADDR]);
                fclose($uh);
            } else { // if cookie is
incorrect
                show_login_page($errmsg);
            }
        } else { // if cookie is not
set
            show_login_page($errmsg);
        }
    }
} else {

```

```

    echo 'Maaf, Anda tidak diperkenankan mengakses berkas ini secara langsung.';
}
?>

```

footer_logged.php

```

<?php
/*-----
Copyright 2009. Anom Tejo Pratomo
Departemen Teknik Mesin - Universitas Indonesia
-----*/
?>

```

```

<center><b><a href="about.html" target="_blank">Tentang</a> | <a href=
"/help_support.html" target="_blank">Bantuan & Dukungan</a> | <a
href="index.php?logout" onclick="unhook();unhook()">Logout</a></b> <script
type="text/javascript" src="js/hitung_mundur.js"></script>
<br><b>i-RoMan 1.0 (beta). &copy;2009, Departemen Teknik Mesin - Universitas
Indonesia</b></center>
</body>
</html>

```

footer_unlogged.php

```

<?php
/*-----
Copyright 2009. Anom Tejo Pratomo
Departemen Teknik Mesin - Universitas Indonesia
-----*/
?>

```

```

<br><center><b><a href="about.html" target="_blank">Tentang</a> | <a href=
"/help_support.html" target="_blank">Bantuan & Dukungan</a>
<br><b>i-RoMan 1.0 (beta). &copy;2009, Departemen Teknik Mesin - Universitas
Indonesia</b></center>
</body>
</html>

```

header.php

```

<?php
/*-----
Copyright 2009. Anom Tejo Pratomo
Departemen Teknik Mesin - Universitas Indonesia
-----*/
?>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
<title>i-RoMan</title>
<meta name="author" content="Departemen Teknik Mesin - Universitas Indonesia">
<meta name="editor" content="Notepad++">
<meta name="description" content="Web-based Robot Manipulation">

```

```

<meta name="keywords" content="web-based, robot, manipulation">
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="pragma" content="no-cache">
<link rel="shortcut icon" href="images/favicon.ico">
</head>
<body style="background-image:url('images/background.jpg');background-attachment:
fixed">
<h1 align="center">i-RoMan</h1>
<table width="360" border="1" align="center" valign="center"><tr>
<td bgcolor="#0099cc" align="center">
<h2>Notifikasi</h2>
</td></tr>
<tr><td bgcolor="#66ccff">
<div style="margin:6px">

```

cek_kecepatan.php

```

<?php
/*-----
Copyright 2009. Anom Tejo Pratomo
Departemen Teknik Mesin - Universitas Indonesia
-----*/
error_reporting(0);

// MEMULAI PENGUJIAN KECEPATAN AKSES INTERNET PENGGUNA
$kb=1024; // Besar data yang akan dikirim untuk
pengujian
//echo "streaming $kb Kb...<!--";
flush();
$time = explode(" ",microtime());
$start = $time[0] + $time[1];
for($x=0;$x<$kb;$x++){
    echo '<!--';
    echo str_pad(" ", 1024, '.'); // Mengirim data, tag "<!--" dan "--" digunakan untuk
menyembunyikan output dari user interface.
    echo '->';
    flush();
}
$time = explode(" ",microtime());
$finish = $time[0] + $time[1];
$deltat = $finish - $start;
$kecepatan = round($kb / $deltat, 3);
//echo "Test finished in $deltat seconds. Your speed is ". round($kb / $deltat, 3)."Kb/s";

// FUNGSI YANG AKTIF KETIKA KECEPATAN AKSES INTERNET PENGGUNA DI BAWAH
KETETAPAN
function limiter(){
    include("templates/header.php"); ?>
    <br>Maaf, bandwidth anda tidak mencukupi untuk menjalankan program
ini.<br><br>
    <script>setTimeout("self.window.location =
'index.php?logout=true'",4000)</script>
    <center><br><a href="index.php?logout=true"><b>Klik di sini jika tidak otomatis
dialihkan!</b></a></center>
    </div></td>

```



```

        </tr></table>
        <br>
        <?php include("templates/footer_unlogged.php"); ?>
    <?php
    }?>
logout.php
    <?php
    /*-----
    Copyright 2009. Anom Tejo Pratomo
    Departemen Teknik Mesin - Universitas Indonesia
    -----*/

    $uh = fopen('kontrol_pengguna/kontrol.txt', 'r') or die ("Berkas tidak dapat dibuka!");
    $ip_tercatat = fread($uh,15);
    fclose($uh);
    // DETEKSI PERMINTAAN LOGOUT
    if (isset($_REQUEST['logout'])) { // Deteksi URL apakah ada variabel "logout"
        if ($ip_tercatat == $_ENV[REMOTE_ADDR]) {
            setcookie('sascookie',"",time()-3600); // Meng-kadaluarsa-kan cookie
            include("templates/header.php");
            echo '<p align="center"><blink><font color="red"><b>Anda ter-logout
secara otomatis!</b></font></blink></p>';
            echo '<center><br><a href="javascript:window.close();"><b>Klik di sini
untuk menutup window!</b></a></center>';
            $uh = fopen('kontrol_pengguna/kontrol.txt', 'w') or die ("Berkas tidak dapat
dibuka!"); // Membuka file "kontrol.txt" dan menulis nilai "0" yang berarti tidak ada sesi yang sedang
aktif
                fwrite($uh, 0);
                fclose($uh);
            } else {
                include("templates/header.php"); ?>
                <p align="justify">Maaf, Anda tidak berhak melakukan eksekusi ini!</p>
                </div></td>
                </tr></table>
                <br>
                <?php include("templates/footer_unlogged.php");
                exit();
            }
        }
    }?>
    </div></td>
    </tr></table>
    <?php include("templates/footer_unlogged.php"); ?>

```


inti.js

```

/*-----
Copyright 2009. Anom Tejo Pratomo & Adhitya Bayu W.
Departemen Teknik Mesin - Universitas Indonesia
-----*/

window.status = "Web-based Robot Manipulation";           // Judul Status Bar

// MENGAKTIFKAN DARURAT
function darurat() {
    menit = 9;penghitung_detik = 0;

window.open("berkas.php?id=eon","", "toolbar=no,menubar=no,location=no,width=500,
height=500");
}

// MENGAKTIFKAN DARURAT
function reset() {
    menit = 9;penghitung_detik = 0;

window.open("berkas.php?id=reset","", "toolbar=no,menubar=no,location=no,width=500
,height=500");
}

// VALIDASI MASUKAN
function validasi(field) {
    var value = field.value;
    if(isNaN(value)) {                                     // Deteksi
        masukan selain angka
        field.style.backgroundColor = 'red';             // Merubah warna form masukan
        menjadi merah sebagai tanda
        alert("Masukkan hanya dapat berupa angka!");   // Alert ke pengguna
        field.value="";                                   // Penghapusan
        masukan selain angka
    } else {
        field.style.backgroundColor = 'white';           // Reset form masukan
    }
}

// PEMROSESAN KOORDINAT
function tangkap_koordinat(e, id) {
    var el = e.target || e.srcElement;
    // Tangkap Koordinat Mouse
    var ePos = {x:0, y:0};
    if ('number' == typeof e.pageX){
        ePos.x = e.pageX;
        ePos.y = e.pageY;
    } else if ('number' == typeof e.clientX){
        ePos.x = e.clientX + document.body.scrollLeft
        + document.documentElement.scrollLeft;
        ePos.y = e.clientY + document.body.scrollTop
        + document.documentElement.scrollTop;

```

```

}
// Tangkap posisi x=0 dan y=0 dari imej
var elPos = {x:0, y:0};
if (el.offsetParent) {
  elPos.x = el.offsetLeft;
  elPos.y = el.offsetTop;
  while (el = el.offsetParent) {
    elPos.x += el.offsetLeft;
    elPos.y += el.offsetTop;
  }
}
// Menampilkan posisi relative kursor mouse terhadap imej
koordinat_x = (ePos.x - elPos.x)-1;
koordinat_y = (ePos.y - elPos.y)-1;
if (id == 'cam_1') {
  document.getElementById('cam_1').value = koordinat_x+', '+koordinat_y; //
Menampilkan koordinat di form untuk Web Camera #1
} else if (id == 'cam_2') {
  document.getElementById('cam_2').value = koordinat_x; //
Menampilkan koordinat di form untuk Web Camera #1
}
}

// MENEMPATKAN VARIABEL-VARIABEL KE BAGIAN PEMROSESAN
function diklik(id) {
  if (id == 'cam_1') {
    document.getElementById("cam_1_x_diklik").value = koordinat_x;
    document.getElementById("cam_1_y_diklik").value = koordinat_y;
  } else if (id == 'cam_2') {
    document.getElementById("cam_2_z_diklik").value = koordinat_x;
  }
}
function konfirmasi(id) {
  konfirmasi_x = document.getElementById("cam_1_x_diklik").value
  konfirmasi_y = document.getElementById("cam_1_y_diklik").value
  konfirmasi_z = document.getElementById("cam_2_z_diklik").value
  konfirmasi_theta_1 = document.getElementById("theta_1").value
  konfirmasi_theta_2 = document.getElementById("theta_2").value
  konfirmasi_theta_3 = document.getElementById("theta_3").value
  konfirmasi_theta_4 = document.getElementById("theta_4").value
  konfirmasi_theta_5 = document.getElementById("theta_5").value
  if (id == 1) {
    if (konfirmasi_x == "" || konfirmasi_y == "" || konfirmasi_z == "") {
      alert("Tidak boleh ada input kosong. Periksa kembali input datanya!")
    } else {
      if (konfirmasi_x > 320 || konfirmasi_y > 240 || konfirmasi_z > 320) {
        alert("Input yang anda masukkan melebihi jangkauan sistem. Silakan
periksa kembali input datanya!")
      } else {
        var answer = confirm("Apakah anda yakin ingin memproses:\n\nx
= "+konfirmasi_x+"; y = "+konfirmasi_y+" dan z = "+konfirmasi_z+"?")
        if (answer){
          menit = 9;penghitung_detik = 0;
          window.open("berkas.php?id="+id+"&x="+konfirmasi_x+"&y="+konfirmasi_y+"&z="+

```



```

window.onunload = function(){
    if (chook) {

window.open("logout.php?logout","logout","toolbar=no,menubar=no,location=no,width=
500,height=500");
    }
}
function unchook() { // Bypass fungsi
    chook=false;
}

// VALIDASI KEYSTROKE DARURAT
function keycheck(event) {
    if(event.keyCode==27) { // Deteksi "Esc" key
        unhook();darurat(); // Menjalankan fungsi darurat
    }
}

webcam.js
function reload1(fps) {
    var now1 = new Date();
    var webcamImg1 = "http://152.118.104.164:7777/out.jpg" + "?" +
now1.getTime();
    document.webcam1.src = webcamImg1;
    setTimeout("reload1("+fps+)", 1000/fps);
}
function reload2(fps) {
    var now2 = new Date();
    var webcamImg2 = "http://152.118.104.164:8888/out.jpg" + "?" +
now2.getTime();
    document.webcam2.src = webcamImg2;
    setTimeout("reload2("+fps+)", 1000/fps);
}

hitung_mundur.js
/*-----
Copyright 2009. Anom Tejo Pratomo
Departemen Teknik Mesin - Universitas Indonesia
-----*/

// KONFIGURASI
var menit = 9; // Lamanya hitung mundur (dalam menit)
var detik = 59; // Detik standar (jangan diubah kecuali Anda tahu
yang Anda lakukan)
var penghitung_detik = detik; // Set variabel detik yang lain untuk dimanipulasi

// HITUNG MUNDUR
penghitung_detik = 0;
function hitung_mundur() {
    penghitung_detik--; // Setiap siklus 1 detik
    mengurangi nilainya 1 poin
    if (penghitung_detik == -1) { // Deteksi detik ketika nilainya "0"

```

```

        menit--; // Setiap siklus 1 menit
    mengurangi nilainya 1 poin
    penghitung_detik = detik; // Me-reset detik untuk memulai
    hitung mundur menit yang baru
    if (menit <= -1) { // Hitung mundur selesai
        menit = 0; penghitung_detik = 0; // Menset menit dan detik ke "0"
        clearTimeout(penghitung); // Stop hitung mundur
        unhook(); unhook(); // Bypass Handler
    }
    Notifikasi "OnUnload" dan Pemaksa Logout
    window.location="index.php?logout" // Otomatis logout jika
    hitung_mundur selesai
}
}
if (document.getElementById) {
    if (penghitung_detik < 10) {
        tampil_penghitung_detik = "0"+penghitung_detik;
    } else {
        tampil_penghitung_detik = penghitung_detik;
    }
    if (menit < 10) {
        tampil_menit = "0"+menit;
    } else {
        tampil_menit = menit;
    }
}

document.getElementById("hitung_mundur_tampil").innerHTML=tampil_menit+":"+tampil_penghitung_detik; // Memasukkan nilai variabel menit dan detik untuk ditampilkan
}
penghitung=setTimeout("hitung_mundur()", 1000); // Set siklus penghitungan mundur
(standar: 1 detik)
}

// INISIALISASI
if (document.all||document.getElementById)
document.write(' - <b id="hitung_mundur_tampil">0 '+menit+':'+penghitung_detik+' </b>'); // Format tampilan hitung mundur di antarmuka
hitung_mundur();

```