

## **Aplikasi *Embedded Internet* pada *Vending Machine* Menggunakan *Microprocessor Rabbit RCM3200***

**Resmana Lim , Lauw Lim Un Tung**

Fakultas Teknologi Industri, Jurusan Teknik Elektro, Universitas Kristen Petra  
e-mail : resmana@petra.ac.id ; tung@petra.ac.id

**Hendrawan**

Alumni Fakultas Teknologi Industri, Jurusan Teknik Elektro, Universitas Kristen Petra  
e-mail : kyut\_one@yahoo.com

### **Abstrak**

Pada paper ini akan dibahas sebuah pengembangan model *Vending Machine* yang dilengkapi dengan koneksi Internet. Mesin dapat dimonitor secara jarak jauh melalui Internet untuk memantau hasil transaksi penjualan serta temperatur mesin pendingin. Sistem *embedded vending machine* terdiri dari 2 bagian, yaitu *vending machine* yang terhubung pada modul RabbitCore yang terkoneksi dengan jaringan Internet dan komputer administrator/kendali sebagai pengendali atau pemonitor. *Software* yang digunakan pada modul RabbitCore adalah Dynamic C sedangkan pada komputer kendali digunakan Apache *Web Server*, PHP dan MySQL *database server*. Pengujian dilakukan dengan cara mengakses mesin ini dari jaringan Internet dengan melakukan pemantauan transaksi penjualan serta pemantauan suhu mesin pendingin. Sistem berjalan baik dan dapat melayani transaksi penjualan yang kesemuanya dapat dimonitor oleh komputer kendali.

**Kata kunci:** *vending machine, microprocessor rabbit, embedded networking, internet.*

### **Abstract**

*The paper describes development of an embedded vending machine equipped with an Internet connection. The machine could be monitored and controlled remotely via Internet to perform temperature monitoring and selling transaction monitoring. This embedded vending machine system consist of two parts, which are an vending machine connected to Rabbitcore module and an administrator/main computer. The software deployed at Rabbitcore module was written in Dynamic C. The administrator computer was running an Apache Web Server, PHP and MySQL database server. The experiment was conducted by connecting the system to the local area network which is connected to the Internet Administrator computer could monitor the up to date information like temperature and selling transaction data. The system was successfully being monitored from the Internet.*

**Keywords :** *vending machine, rabbit microprocessor, embedded networking, internet.*

### **1. Pendahuluan**

*Vending Machine* adalah sebuah mesin penjual minuman yang dapat beroperasi secara *standalone* untuk melayani transaksi pembelian minuman atau makanan kecil. *Vending Machine* konvensional yang beredar saat ini memiliki beberapa kekurangan yaitu pengelola tidak dapat mempunyai informasi secara on-line yang sewaktu-waktu bisa diakses misalnya tentang kondisi mesin seperti: hasil transaksi penjualan, stok produk yang tersimpan pada mesin, kondisi mesin pendingin dan lain-lain.

Untuk mengatasi masalah itu pada penelitian ini dibangun sebuah model sistem *embedded* pengendali *vending machine* yang memiliki koneksi Internet (*embedded Internet*). Dengan demikian seluruh aktifitas mesin dapat dimonitor oleh pengelola mesin secara jarak jauh sepanjang masih ada koneksi web Internet. Pengelola dapat memonitor transaksi penjualan, stok produk serta temperatur mesin pendingin. Sistem *embedded Internet* saat ini sudah mulai populer diaplikasikan pada sistem kendali, peralatan rumah tangga, instrumentasi dan lain-lain [1,2].

Selanjutnya paper ini diorganisasi sebagai berikut: bagian 2 mengulas tentang teknologi *embedded internet* yang digunakan, bagian 3 membicarakan tentang perancangan sistem dan bagian 4 tentang pengujian sistem serta diakhiri dengan penutup.

---

**Catatan:** Diskusi untuk makalah ini diterima sebelum tanggal 1 Desember 2004. Diskusi yang layak muat akan diterbitkan pada Jurnal Teknik Elektro volume 5, nomor 1, Maret 2005.

## 2. *Embedded Internet Rabbitcore*

### 2.1 *Mikroprosesor Rabbit 3000* [3,4]

Mikroprosesor yang akan digunakan adalah mikroprosesor Rabbit 3000 mempunyai kemampuan yang tinggi, *low-EMI microprocessor* yang didesain secara spesifik untuk *embedded control*, komunikasi, dan koneksi dengan *Ethernet* dan Internet.



Gambar 1. Modul RCM3200 [1]

*Buffer I/O* Rabbit mempunyai batas kemampuan arus *sourcing* dan *sinking* sebesar 6.8 mA per *pin* dengan batas tegangan maksimum untuk semua I/O adalah 5.5 V.

RCM3200 mempunyai I/O *parallel* sebanyak 52 buah yang dikelompokkan menjadi 7 *port* (masing-masing 8 bit) yang tersedia pada *header* J1 dan J2. Sebanyak 44 *bidirectional I/O* terletak pada *pin* PA0-PA7, PB0, PB2-PB7, PD2-PD7, PE0-PE1, PE3-PE7, PF0-PF7, dan PG0-PG7.

### 2.2 *Dynamic C* [5,6]

*Dynamic C* merupakan *integrated development system* yang dirancang khusus untuk *embedded software* (yang menggunakan bahasa pemrograman C). *Software* ini didesain untuk digunakan pada kontroler Z-World dan kontroler lainnya yang berbasis mikroprosesor Rabbit. Rabbit 2000 dan Rabbit 3000 merupakan mikroprosesor 8 bit yang mempunyai performa yang tinggi yang dapat menerima aplikasi bahasa C sampai dengan 50.000 *statements* atau 1 MB.

*Dynamic C* terintegrasi dari beberapa fungsi yaitu *editing*, *compiling*, *linking*, *loading*, dan *debugging* dalam 1 program. Pada kenyataannya, *compiling*, *linking*, *loading* adalah 1 fungsi. Di dalam *Dynamic C* terdapat *editor* teks yang mudah untuk digunakan. Program dapat dieksekusi dan *debugged* secara interaktif pada level kode mesin. Menu-menu dan *keyboard*

*shortcut* tersedia juga dalam *Dynamic C*. Selain itu, bahasa *assembly* juga didukung oleh *Dynamic C* dan dapat digunakan bersama-sama dengan bahasa C.

Cara memprogram mikroprosesor menggunakan *Dynamic C* adalah sbb :

1. Membuat *listing* program (baik dalam bahasa C atau *assembly*),
2. Langkah selanjutnya program di-*compile* agar dapat di-*download* pada mikroprosesor dengan menekan tombol F9.

### 2.3 Pemrograman *Socket* dengan PHP

Penelitian ini mencoba untuk mengaplikasikan PHP terutama penggunaan pemrograman *socket*. *Socket* menyediakan jalan bagi *client* dan *server* untuk saling berkomunikasi di dalam jaringan komputer. *Socket* membuat jalur *end to end communication* (yang memungkinkan *client* untuk mengirim *request* ke *server* dan *server* menerima *request* tersebut dan memberikan respon kepada *client*). Sebagai contoh adalah sebuah *web server* melayani *request client* dengan membuka sebuah *socket* (biasanya *port* 80) dan *client* (*web browser*) berkomunikasi dengannya melalui *socket* tersebut. Untuk dapat mengaktifkan fitur *socket* pada PHP, maka perlu menghilangkan tanda “;” pada baris *extension=php\_sockets.dll* dalam *file* ‘*php.ini*’.

Beberapa fungsi yang dipakai adalah sebagai berikut :

- *Socket\_create()* untuk membentuk *socket*.  
`resource socket_create (int domain, int type, int protocol)`
- *Socket\_bind()* untuk memberi nama pada *socket* (mempunyai nilai balik, **TRUE** saat sukses atau **FALSE** saat gagal).  
`bool socket_bind (resource socket, string address [, int port])`
- *Socket\_listen()* untuk mendengar koneksi pada *socket* (mempunyai nilai balik **TRUE** saat sukses atau **FALSE** saat gagal).  
`bool socket_listen (resource socket [, int backlog])`
- *Socket\_accept()* untuk menerima koneksi pada *socket*.  
`resource socket_accept (resource socket)`
- *Socket\_close()* untuk menutup *socket*.  
`void socket_close (resource socket)`
- *Socket\_read()* untuk membaca *bytes* yang ada pada *socket*.

```
string socket_read (resource
socket, int length [,int type])
• Socket_write() untuk menulis pada socket.
int socket_write (resource socket,
string buffer [, int length])
```

Contoh pemrograman *socket* dengan PHP :

```
<?
$host = "192.168.12.13";
$port = 1241;
// bentuk socket
$socket = socket_create(AF_INET,
SOCK_STREAM, 0) or die("Socket gagal
dibentuk\n");
// bind socket pada port
$result = socket_bind($socket, $host,
$port) or die("Socket gagal di-
bind\n");
// mulai listening
$result = socket_listen($socket, 3)
or die("Listen Gagal\n");
// terima request koneksi
// bentuk socket lain untuk meng-
handle komunikasi
while(1)
{
$spawn = socket_accept($socket) or
die("Gagal menerima koneksi\n");
// baca input dari client
$input = socket_read($spawn, 1024) or
die("Gagal membaca input\n");
echo $input;
// kirim kembali ke client
$output = "alo\n";
socket_write($spawn, $output, strlen
($output)) or die("Gagal membuat
output\n");
// tutup socket
socket_close($spawn);
}
socket_close($socket);
?>
```

Listing diatas menjadikan komputer dengan IP 192.168.12.13 sebagai *server* yang membuka layanan pada *port* 1241 dan setelah menerima *input* dari *client*, *server* akan mengirimkan balasan pada *client*.

### 3. Perencanaan Sistem



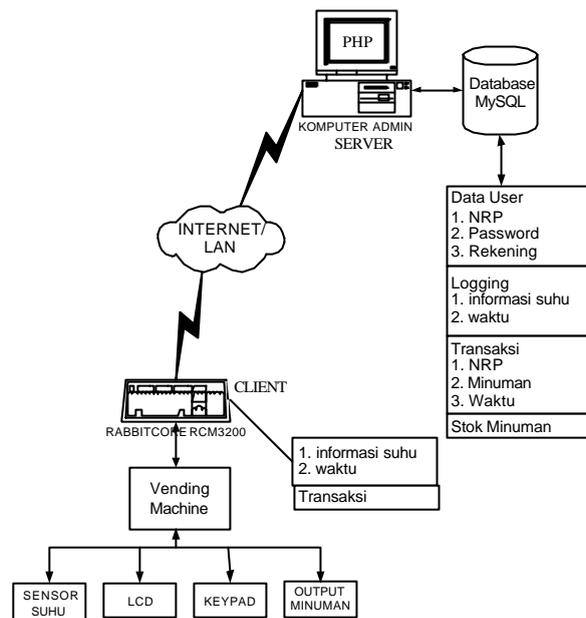
Gambar 2. *Vending Machine*

### 3.1 Perencanaan Sistem *Vending Machine*

Sistem ini dibagi menjadi 2 bagian utama yaitu modul Rabbitcore RCM3200 dan komputer *administrator*.

Modul Rabbitcore RCM3200 diaplikasikan sebagai *vending machine* dan terdapat *interface input* dan *output* yaitu :

- *Driver Solenoid*
- *Keypad*
- *LCD 16x2*
- *Sensor Suhu*



Gambar 3. Sistem *Embedded Vending Machine*

Tabel 1. Alokasi *Port* Modul Rabbitcore RCM3200

Interface	Port Rabbit	Input	Output
Driver Solenoid	Port F	Tidak ada	4 bit (PF0 –PF3)
Keypad	Port E	5 bit (PE3 – PE7)	Tidak ada
LCD 16x2 (mode 4 bit)	Port A, D	Tidak ada	6 bit (PA4 –PA7; PD2 – PD3)
Sensor Suhu	Port D, G	3 bit (PD5; PD6- PD7)	8 bit (PG0 –PG7)

### 3.2 Perencanaan *Software*

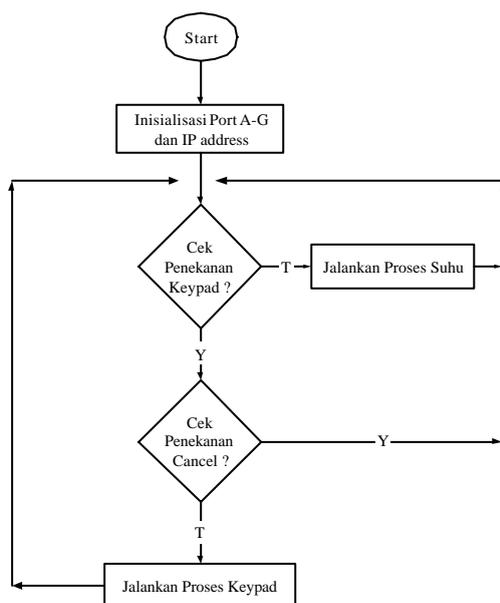
Dalam perencanaan *software* dibedakan menjadi dua bagian utama, yaitu *software* untuk modul RabbitCore RCM3200 (*vending machine*) dan *software* pada komputer *administrator*. *Vending machine* berbasis RabitCore disebut pula

sebagai komputer client sedangkan computer administrator disebut pula sebagai server. Administrator/Server berfungsi untuk pencatatan data-data vending machine dan layanan akses pengguna,

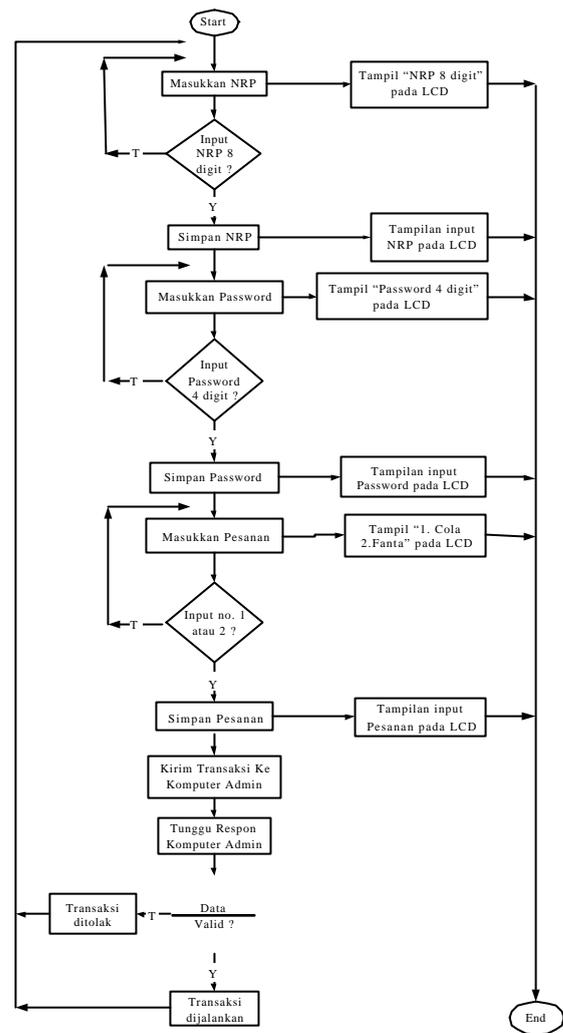
### 3.2.1. Software RabbitCore RCM3200 (*Vending Machine*)

Cara kerja dari RabbitCore adalah saat pertama kali dinyalakan, maka akan dilakukan inialisasi *port* dan alamat IP tertentu. Kemudian mulai mencek penekanan *keypad*. Jika tidak ada penekanan *keypad*, maka program akan menunggu selama 60 detik. Dan setelah waktu 60 detik terpenuhi tanpa ada penekanan *keypad*, maka dijalankan pengecekan *input* data (berupa suhu). Dan jika data telah siap atau tersedia, data tersebut dikonversikan menjadi derajat celsius yang kemudian dikirimkan pada *server*. Kemudian dilakukan cek penekanan *keypad* kembali.

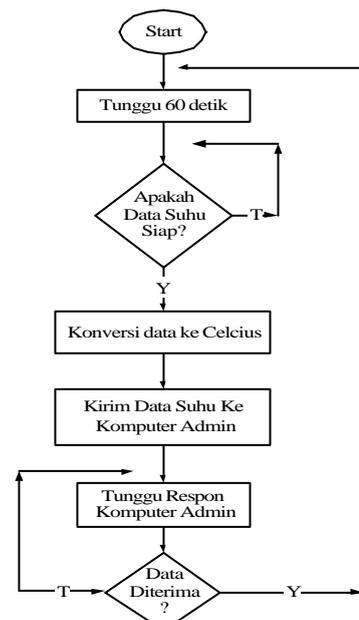
Jika ada penekanan *keypad*, maka RabbitCore akan meminta *input* data berupa NRP, *password* dan pilihan minuman yang dipesan serta menampilkannya pada LCD. Dan setelah semua *input* data masuk, maka data tersebut akan dikirimkan pada *server* (untuk dicek validasinya). Setelah mendapat balasan dari *server* (bahwa data tersebut valid), maka transaksi pembelian dijalankan. Dan jika tidak valid, maka transaksi ditolak. Untuk membatalkan transaksi, *user* dapat menekan tombol "cancel" pada *keypad*.



Gambar 4. Algoritma Sistem *Vending Machine*



Gambar 5. Algoritma Proses *Keypad*



Gambar 6. Algoritma Proses Suhu

3.2.2. Software pada Komputer Administrator

3.2.2.1. Database MySQL

Pada komputer *administrator*, dibuat sebuah *database* dengan nama *vending* yang mempunyai 4 tabel, yaitu :

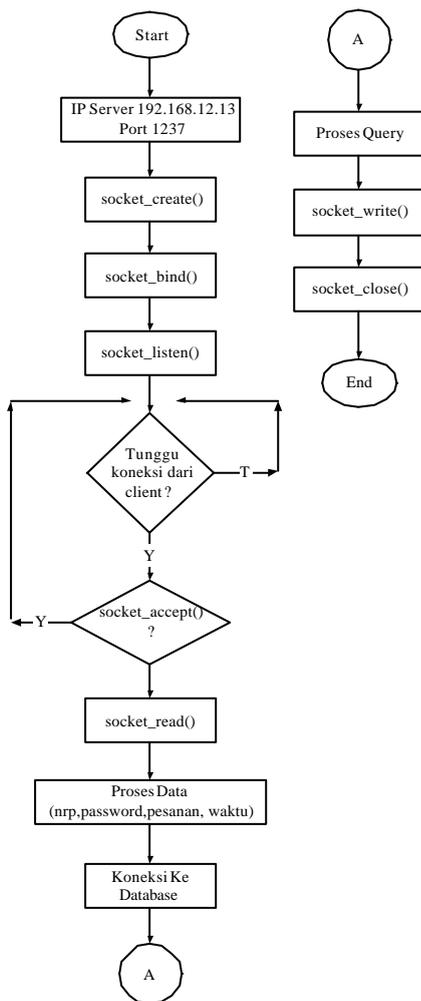
- Tabel **data\_user** digunakan untuk menyimpan data-data penting pelanggan atau konsumen. Kolom saldo akan berkurang nilainya, jika terjadi transaksi pada pelanggan yang bersangkutan sesuai dengan harga yang telah ditentukan.
- Tabel **logging** digunakan untuk menerima dan menyimpan data berupa suhu dan waktu yang dikirim oleh *vending machine*.
- Tabel **stok\_minuman** digunakan untuk menyimpan jumlah stok minuman yang tersedia pada *vending machine*.
- Tabel **transaksi** digunakan untuk menerima dan menyimpan transaksi yang terjadi.
- Tabel **voucher** digunakan untuk menyimpan pembelian *voucher*.

3.2.2.2. PHP

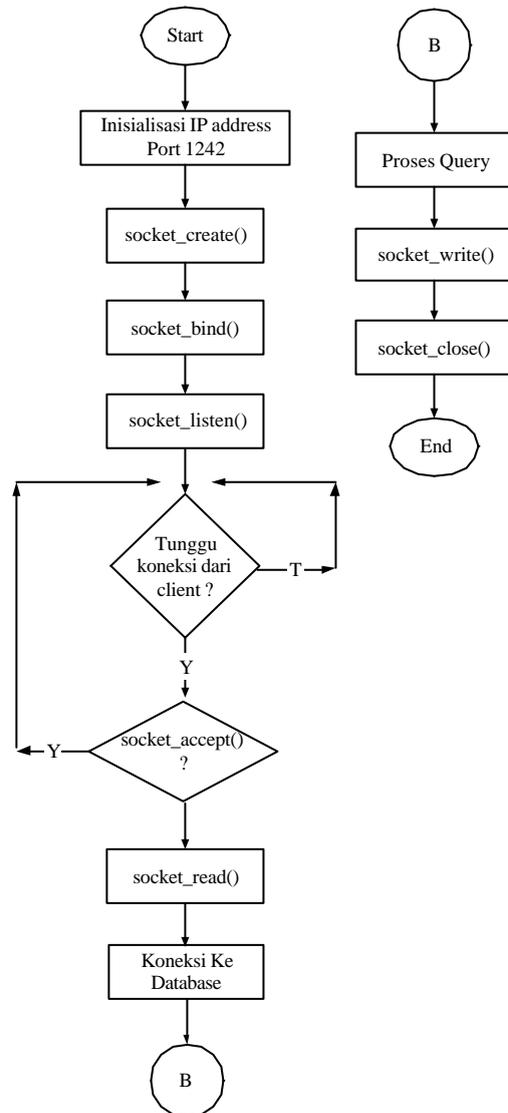
Terdapat 2 skrip program utama yaitu 'server\_NRP\_password.php' dan 'server\_logging.php'.

File 'server\_NRP\_password.php' digunakan agar komputer *admin* menjadi *server* pada *port* 1237 (yang menunggu, menerima *input* dari *vending machine* berupa data NRP, *password* dan pilihan minuman, serta memberi balasan berupa validasi transaksi).

File 'server\_logging.php' menjalankan komputer *admin* menjadi *server* pada *port* 1242 (yang menunggu, menerima *input* dari *vending machine* berupa data suhu dan waktu, kemudian menyimpannya pada *database*, serta memberi balasan pada *vending machine* bahwa data telah diterima).



Gambar 7. Algoritma Server\_NRP\_password.php



Gambar 8. Algoritma Server\_logging.php

## 4. Pengujian Sistem

### 4.1 Pengujian Jaringan

Pengujian jaringan dilakukan pada sebuah jaringan LAN. *Vending machine/client* beserta dengan komputer administrator/server dikoneksikan dengan menggunakan ethernet switch. Pengujian awal dilakukan dengan cara melakukan test koneksi menggunakan ‘ping’ antara komputer client dan server.

### 4.2 Kondisi Awal

#### 4.2.1 Kondisi Awal *Vending Machine*

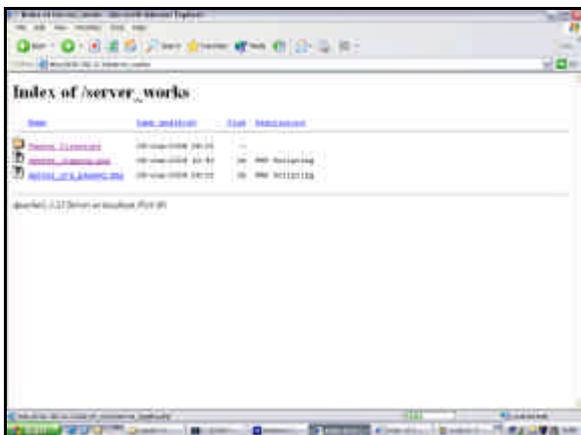
Pada saat *start* awal, *RabbitCore* akan melakukan inialisasi *port* dan alamat IP *embedded vending machine* kemudian ucapan selamat datang yang ditampilkan pada LCD. Tampilan akan terus dipertahankan selama tidak ada penekanan *keypad*.

#### 4.2.2 Kondisi Awal Komputer Admin

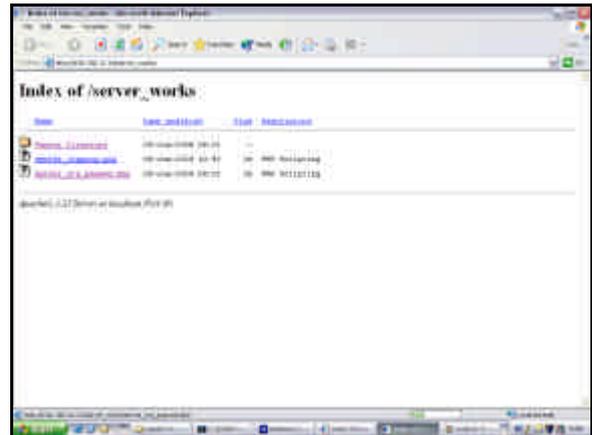
##### 4.2.2.1 Menjalankan Skrip PHP Pada *Web Server*

Untuk menjadikan komputer *admin* sebagai *server*, maka dijalankan *file* ‘*server\_logging.php*’ dan ‘*server\_nrp\_password.php*’ melalui *web browser* seperti Internet Explorer dengan menuliskan alamat URL [http://192.168.12.13/server\\_works/server\\_logging.php](http://192.168.12.13/server_works/server_logging.php) dan [http://192.168.12.13/server\\_works/server\\_nrp\\_password.php](http://192.168.12.13/server_works/server_nrp_password.php).

*File* ‘*server\_logging.php*’ berfungsi untuk membuka *port* 1242 yang ditujukan untuk koneksi terhadap data suhu dari *client* dan *file* ‘*server\_nrp\_password.php*’ membuka *port* 1237 yang ditujukan untuk koneksi dengan *client* dalam pertukaran data transaksi.



Gambar 9. Tampilan *File* *Server\_logging.php* Dijalankan



Gambar 10. Tampilan *File* *Server\_nrp\_password.php* Dijalankan

##### 4.2.2.2 Kondisi Awal *Database MySQL*

Pada pengujian awal *database*, untuk melihat data pada tabel-tabel yang terdapat dalam *database* “*vending*” dapat diakses melalui *web browser* dengan URL [http://192.168.12.13/server\\_works/mysql/tampil.php](http://192.168.12.13/server_works/mysql/tampil.php).



Gambar 11. Tampilan Awal Tabel-tabel Pada *Database Vending*

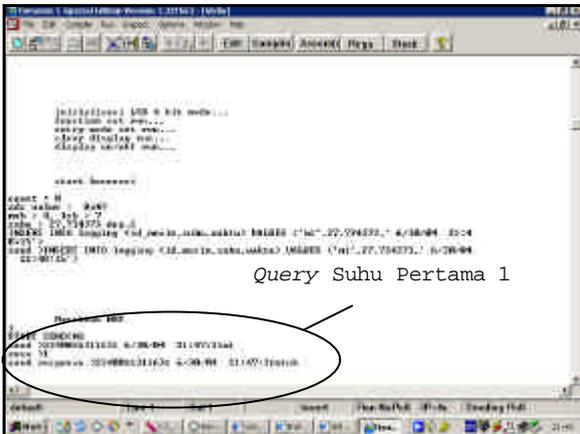
Dengan dilakukan pengujian pada kondisi awal komputer *admin* dan *vending machine*, membuktikan bahwa *vending machine* dan komputer *admin* telah berhasil berjalan sesuai dengan perencanaan sistem.

### 4.3 Kondisi Pengujian Sistem

#### 4.3.1 Proses *Monitoring* Suhu

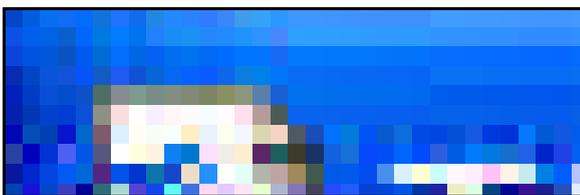
Jika selama 60 detik tidak ada penekanan, maka akan dijalankan proses *sampling* suhu. Dan setelah data siap, maka *vending machine* akan mengirimkan *query* pada komputer *server*.

Pengamatan proses *monitoring* suhu pada *vending machine* menggunakan PC *console* seperti yang terlihat pada gambar berikut.



Gambar 12. Tampilan *Monitoring* Suhu Pertama

Gambar 12. merupakan hasil pengamatan proses *monitoring* suhu pada menit pertama sejak program dijalankan. Setelah *sampling* suhu didapat, maka hasil dari *sampling* suhu tersebut (yang berupa nilai derajat celcius) akan dikirimkan ke *server* dalam bentuk *query*. Selanjutnya oleh *server*, *query* tersebut akan dieksekusi ke dalam *database* “*vending*”. Masing-masing *query* dari proses *sampling* suhu dikirim ke *server*. Oleh *server* kemudian dijalankan dan *database* MySQL akan menyimpan data yang di-*input*-kan (yaitu suhu dan waktu) pada tabel “*logging*”. Perubahan data pada tabel “*logging*” dapat diamati dengan mengakses URL [http://192.168.12.13/server\\_works/mysql/tabel\\_logging.php](http://192.168.12.13/server_works/mysql/tabel_logging.php).



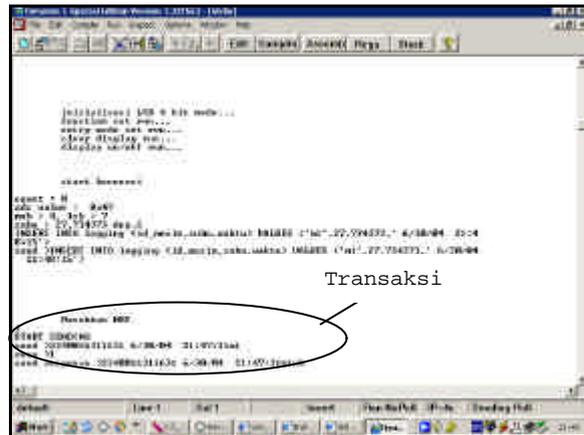
Gambar 13. Perubahan Data Pada Tabel *Logging*

Dengan pengujian ini, maka dapat dibuktikan bahwa proses *monitoring* suhu dapat terlaksana. Selain itu juga membuktikan bahwa komputer *admin* telah berhasil menyimpan data pada tabel “*logging*”.

#### 4.3.2 Proses Transaksi

Proses transaksi dimulai saat ada penekanan *keypad*. Pada proses transaksi, *user* diminta untuk memasukkan NRP sebanyak 8 angka.

Kemudian *password* sebanyak 4 angka. Dan setelah penekanan karakter keempat dari *password* akan muncul pilihan minuman yang akan dipesan, *user* cukup menekan 1 angka yang sesuai dengan minuman yang dipesan. Setelah semua *input* data yang dibutuhkan telah dimasukkan, maka *vending machine* akan menggabungkan ketiga data tersebut menjadi satu berikut waktu transaksinya dan dikirimkan pada komputer *server*.



Gambar 14. Tampilan *Monitoring* Proses Transaksi

Pada proses transaksi, setelah *user* memasukkan semua *input* maka seperti pada gambar 14, semua data *input* tersebut (yaitu *nrp*, *password*, pilihan, waktu transaksi dan *id\_mesin*) akan dikirimkan ke *server*. Dan *server* akan memberikan respon. Sehingga *client* menerima respon dari *server* berupa karakter “1” yang menandakan transaksi valid dan permintaan minuman Coca Cola segera dijalankan.

##### 4.3.2.1 Transaksi Valid

Proses transaksi dinyatakan valid, jika NRP dan *password* yang dimasukkan sesuai dengan data yang tersimpan pada *database server* dan akan diterima balasan dari *server* berupa karakter pilihan minuman yang dipesan. Dan selanjutnya *vending machine* akan menggerakkan mekanik *solenoid*. Sehingga minuman yang dipesan dapat keluar dan *vending machine* akan menampilkan pesan pada LCD agar *user* mengambil minuman yang dipesan tersebut.

Saat proses transaksi dinyatakan valid, maka *server* akan menyimpan transaksi yang terjadi pada tabel “*transaksi*”, meng-*update field* “*rekening*” milik *user* hendrawan pada tabel “*data\_user*” dan mengurangi jumlah minuman

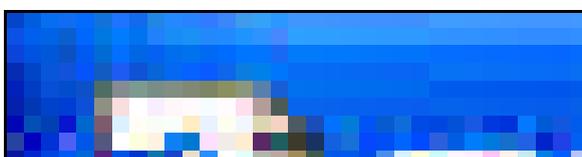
sesuai dengan *record* “id\_mesin” pada tabel “stok\_minuman”. Untuk dapat mengamati tabel “transaksi”, kita dapat mengakses alamat URL [http://192.168.12.13/server\\_works/mysql/tabel\\_transaksi.php](http://192.168.12.13/server_works/mysql/tabel_transaksi.php) dan untuk mengamati tabel “data\_user” dilakukan akses pada alamat URL [http://192.168.12.13/server\\_works/mysql/tabel\\_datauser.php](http://192.168.12.13/server_works/mysql/tabel_datauser.php). Dan tabel “stok\_minuman” pada alamat URL [http://192.168.12.13/server\\_works/mysql/tabel\\_stok.php](http://192.168.12.13/server_works/mysql/tabel_stok.php)



Gambar 15. Penambahan Data Pada Tabel Transaksi



Gambar 16. Perubahan Data Pada Tabel Data\_user Setelah Di-update



Gambar 17. Perubahan Data Pada Tabel Stok\_minuman

#### 4.3.2.2 Stok Minuman Habis

Proses stok minuman habis terjadi apabila transaksi dinyatakan valid namun stok minuman pada tabel “stok\_minuman” samadengan nol atau habis. *Server* akan tetap melakukan penyimpanan data pada tabel “transaksi” namun tidak menjalankan *update* (atau pengurangan rekening *user*) pada tabel “data\_user”. Kemudian *server* mengirimkan karakter “5” atau “6” yang menandakan bahwa minuman yang dipesan telah habis pada *client*.

Untuk dapat mengamati tabel “stok\_minuman”, kita dapat mengakses alamat URL [http://192.168.12.13/server\\_works/mysql/tabel\\_stok.php](http://192.168.12.13/server_works/mysql/tabel_stok.php). Dan apabila salah satu stok minuman habis, maka akan muncul pemberitahuan pada *admin* untuk segera mengisi ulang minuman

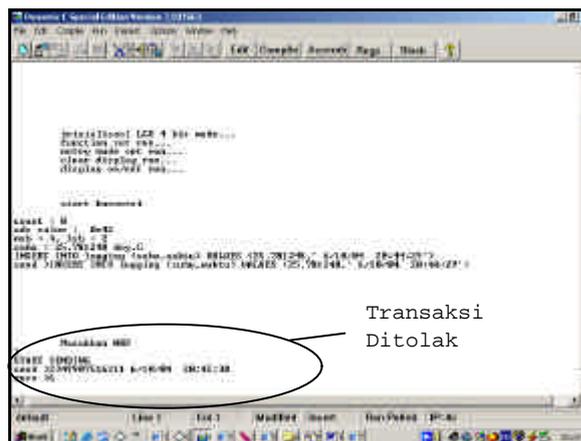
pada *vending machine*. Kemudian *admin* meng-*update* stok minuman pada tabel dengan mengklik link “Update Stok”.



Gambar 18. Tampilan Tabel Stok\_minuman Saat Minuman Habis

#### 4.3.2.3 Transaksi Ditolak

Proses transaksi dinyatakan tidak valid atau ditolak, apabila NRP dan *password* yang dimasukkan tidak sesuai dengan data yang tersimpan pada *database server*. Dan *server* mengirimkan kembali ke *client* berupa string “6”. Dan selanjutnya *vending machine* akan menampilkan pesan pada LCD “Transaksi Ditolak”.



Gambar 19. Tampilan *Monitoring* Proses Transaksi Tidak Valid

Gambar di atas menunjukkan bahwa *client* menerima karakter “6” dari *server* yang menandakan bahwa transaksi *user* ditolak.

#### 4.3.2.4 Pembatalan Transaksi

Proses ini dilakukan oleh *user* apabila terjadi kesalahan dalam memasukkan *input* melalui *keypad* dengan cara menekan tombol “C” (yang berarti *cancel*). Saat tombol ini ditekan, maka *vending machine* akan kembali pada kondisi awal dan *user* dapat melakukan transaksi kembali.

Dengan dilakukannya pengujian ini, membuktikan bahwa *vending machine* ini telah bekerja

dengan baik untuk menjalankan proses transaksi dan komunikasi dengan *server*. Dan komputer *admin* telah bekerja sesuai dengan fungsinya yaitu untuk menerima koneksi (*request* dan respon) dengan *client*, pengolahan data, melakukan koneksi dan menjalankan *query* pada *database*.

#### 4.3.3 Bantrol Antara Proses *Monitoring* Suhu dengan Proses Transaksi

Ada 2 proses yang dijalankan secara bersamaan pada vending machine, yaitu proses *monitoring* suhu dan proses transaksi. Selama proses suhu berjalan, maka vending machine akan menunggu selama 60 detik dan mencek penekanan *keypad*. Di dalam periode 60 detik itu, jika terdapat penekanan *keypad* maka perhitungan periode dihentikan sementara (selama ada penekanan *keypad*). Ketika *user* berhenti sejenak untuk melanjutkan penekanan *keypad*, maka akan dijalankan kembali perhitungan periode yang tersisa. Pada saat periode habis dan bersamaan dengan dilakukan penekanan *keypad* kembali, maka program akan menjalankan proses *monitoring* suhu dan *user* harus melakukan transaksi ulang dengan menekan tombol "cancel".

Dengan dilakukannya pengujian ini, maka dapat disimpulkan bahwa sistem vending machine masih terdapat *bug*, yaitu pada saat terjadi bantrol antara proses *monitoring* suhu dan proses transaksi.

## 5. Kesimpulan dan Saran

### 5.1 Kesimpulan

Berdasarkan hasil perencanaan sistem dan pengujian *embedded vending machine*, maka dapat diambil beberapa kesimpulan :

- Modul yang terhubung pada vending machine telah berfungsi dengan baik sehingga mendukung penerapan aplikasi vending machine konvensional pada umumnya.
- Pengujian *embedded vending machine* telah dapat memecahkan masalah yang dihadapi pada vending machine konvensional di dalam hal mendapatkan informasi mesin yang *up to date* seperti suhu, stok minuman dan waktu terjadinya transaksi.
- Penggunaan modul Rabbitcore RCM3200 yang berbasis mikroprosesor Rabbit 3000 dapat berjalan *standalone* dan sangat

mendukung penerapan pengintegrasian vending machine pada jaringan internet terutama dalam pertukaran data yang dilakukan *client* (vending machine) dan *server* (komputer *admin*).

- Sistem masih belum menjamin sekuritas pengiriman data. Karena data yang dikirim masih dalam bentuk *plain text*, sehingga data dapat dengan mudah untuk dibaca oleh semua orang sehingga masih dapat terjadi kecurangan.
- Kekurangan pada protokol transaksi sistem, yaitu saat itu transaksi yang terjadi belum selesai (contoh: sudah dilakukan pengurangan rekening pada *database*, kemudian pada saat *server* mengirim respon, jaringan internet putus) telah dapat diatasi dengan menambahkan *acknowledgment* pada *client* dan kemudian dilakukan *update* pada *database server*.
- Sistem vending machine masih terdapat *bug*, yaitu pada saat terjadi bantrol antara proses *monitoring* suhu dan proses transaksi. Hal ini diatasi dengan melakukan pembatalan transaksi oleh *user*.

### 5.2 Saran pengembangan lanjutan

- Untuk menghindari terjadinya gangguan pemadaman listrik, vending machine dapat dihubungkan dengan UPS agar tetap dapat melakukan fungsinya.
- Karena internet merupakan media komunikasi yang bersifat publik maka perlu diperhatikan masalah sekuritas data agar dapat mengurangi kecurangan. Sebagai contoh sederhana yaitu dengan melakukan enkripsi data sebelum dikirim melalui internet. Dan pada sisi penerima, data yang terenkripsi tersebut didekripsi sehingga didapatkan data yang sesungguhnya.

### Daftar Pustaka

- [1] Edward Cheung, *A Winning Combination—PIC Internet Connectivity*, Circuit Cellar Magazine Issue 123, October 2000, page 12.
- [2] Alan Singer, *Internet Connectivity—Do-it-Yourself or Off-the-Shelf ?*, Circuit Cellar Magazine Issue 123, October 2000, page 28.
- [3] Rabbit Core RCM3200 Getting Started, Z-World, Inc., 2002. [<http://www.zworld.com/documentation/docs/manuals/RCM3200/GettingStarted/RC3200GS.pdf>]

- [4] Rabbit 3000 Microprocessor User's Manual, Rabbit Semiconductor, 2002. [<http://www.zworld.com/documentation/docs/manuals/Rabbit3000/UsersManual/R3000UM.pdf>]
- [5] Dynamic C User's Manual, Z-World, Inc., 2000. [<http://www.zworld.com/documentation/docs/manuals/DC/DCUserManual/DCPUM.pdf>]
- [6] Dynamic C TCP/IP User's Manual, Z-World, Inc., 2000. [<http://www.zworld.com/documentation/docs/manuals/TCPIP/UsersManual/tcpipuser.pdf>]
- [7] Rabbit Semiconductor Tech Bulletin Board, Z-World, Inc., [<http://www.zworld.com/support/bb/messages/board-topics.html>]