

3. DESAIN SISTEM

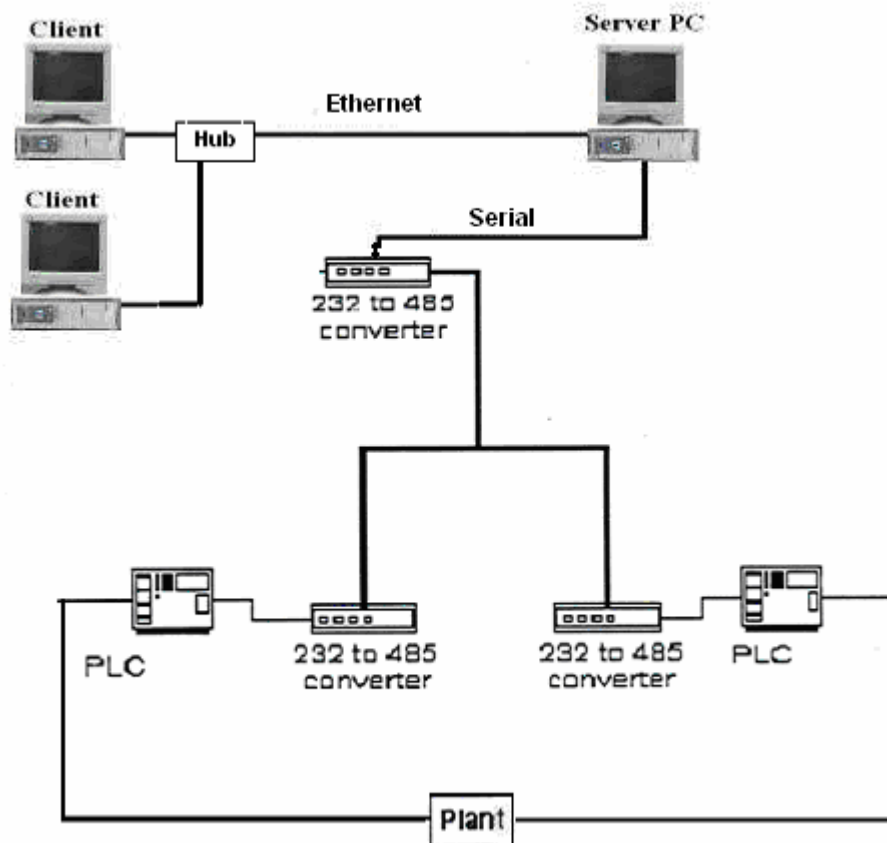
3.1. Deskripsi Sistem

Sistem yang dibuat merupakan penerapan SCADA sistem *Distributed Application* pada Wonderware InTouch 9.0 yang mampu memonitor dan mengontrol perangkat-perangkat pada sebuah miniatur kompleks perumahan dengan menggunakan jaringan komputer dan PLC.

Untuk penerapan SCADA pada saat ini koneksi antar komputer umumnya menggunakan Ethernet (*local area network*). *Network Architecture* yang digunakan pada Tugas Akhir ini adalah Network Application Development (NAD), rekomendasi PLC yang digunakan sehingga arsitektur ini dapat berjalan secara optimum menggunakan PLC yang mempunyai koneksi langsung ke network menggunakan Ethernet seperti :

- PLC OMRON CJ1M-ETN Series
- PLC Schneider Twido 40 I/O LC** 40DRF
- PLC Schneider M340
- PLC Twido + TWD Comm module 499TWD01100, dsb.

Namun karena keterbatasan PLC yang ada adalah OMRON CPM1 yang hanya dapat menggunakan komunikasi serial, maka PLC dihubungkan hanya ke Server PC dan arsitektur NAD tidak dapat berjalan secara optimal karena *Client* tidak dapat mengakses PLC secara langsung. Dengan demikian fungsi arsitektur yang digunakan menjadi mirip dengan *Server-Based Architecture*.



Gambar 3.1. Blok diagram sistem

Pada *plant* sistem keamanan kompleks perumahan ini mempunyai input output PLC yang meliputi *switch push button* pada setiap rumah, *stepper motor*, *inductive proximity sensor* sebagai sensor *gate* masuk dan *gate* keluar, *buzzer* sebagai *alarm*, dan LED sebagai lampu jalan.

Selain itu untuk koneksi komputer menuju PLC OMRON CPM1 dapat digunakan koneksi serial. Untuk berkomunikasi secara serial, PC menggunakan RS 232. Karena RS 232 tidak dapat di-paralel ke beberapa *device*, maka digunakan RS 485 agar PC dapat terhubung ke beberapa *device* secara paralel. Oleh sebab itu, dibutuhkan suatu *converter* dari RS 232 ke RS 485 agar PC dapat berkomunikasi dengan beberapa *device* secara paralel. Untuk lebih jelasnya dapat dibaca di buku TA milik Sdr. Edison Lao (Tugas Akhir : No.02010837/ELK/2007).

Mula-mula *gate* dapat terbuka dan ditutup secara leluasa menggunakan *Human Machine Interface* (HMI) di PC. Jika suatu saat salah satu rumah menekan tombol *emergency* (tanda bahaya), maka *alarm* yang berada di pos keamanan

otomatis berbunyi, kemudian *gate* pintu keluar dan masuk secara otomatis tertutup dan di layar komputer akan menunjukkan rumah mana yang membutuhkan bantuan agar petugas keamanan dengan mudah memberikan pertolongan. *Gate* akan dibuka melalui PC atau dengan saklar jika keadaan sudah terkendali. Saat malam hari lampu jalan dapat menyala otomatis ketika jam menunjukkan pukul 18.00 dan akan mati otomatis pada pukul 06.00, selain itu dapat juga dinyalakan manual melalui saklar atau melalui PC. *Plant* ini menggunakan *supply* 24 VDC dan 5 VDC. Berikut adalah gambar *plant*:



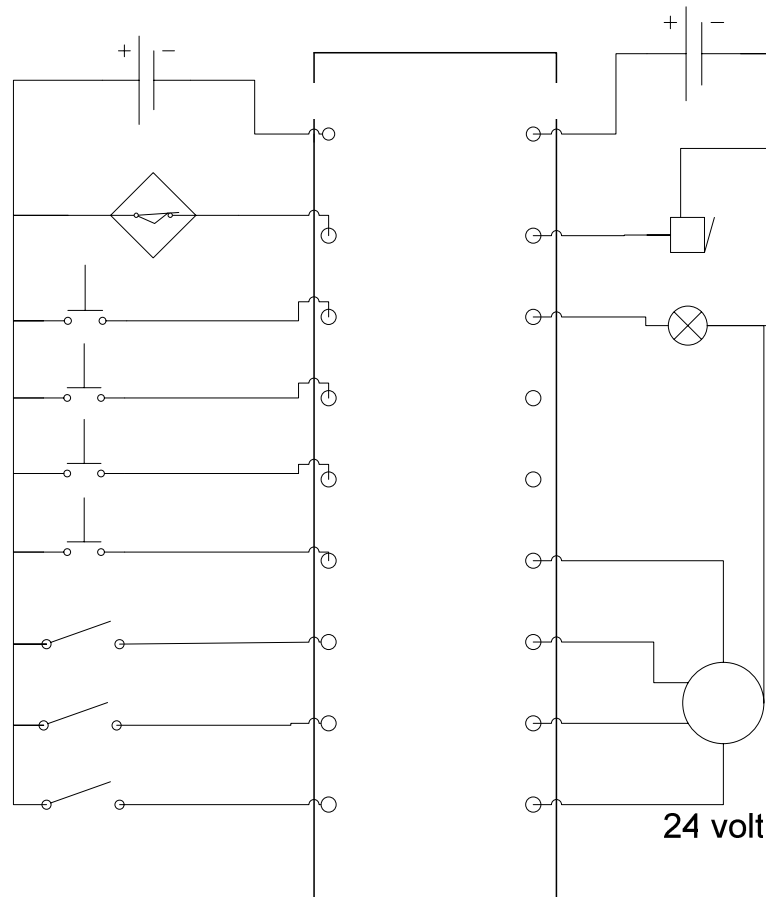
Gambar 3.2. *Plant* kompleks perumahan sederhana

Sumber : Lao, Edison. *Pembuatan Man Machine Interface Pada Jaringan PLC Omron CPM1 Untuk Sistem Keamanan Miniatur Kompleks Perumahan*. Tugas Akhir NO:02010837/ELK/2007. Surabaya: Universitas Kristen Petra, 2007, p.29

3.2. Konfigurasi *Wiring* PLC

PLC yang akan digunakan adalah PLC dari *vendor* OMRON dengan tipe CPM1 yang memiliki 12 *input* dan 8 *output*. Bagian *common* pada pin *input* dihubungkan dengan *ground* power supply yang berarti *input sinking module* dan bagian *common* pada pin *output* dihubungkan pada VCC *power supply* yang berarti *Output Sourcing module*. *Sinking Module* berarti PLC menerima arus dari *input device*, *Sourcing module* berarti PLC yang memberikan arus ke device, *Sinking* berarti PLC menerima arus dari device. Sedangkan Pin *input* tiap PLC digunakan sebanyak 7 (tujuh) pin. Empat pin *input* di tiap PLC digunakan untuk

inputan dari Tombol *emergency*, 1 (satu) untuk tombol saklar lampu jalan, dan 2 (dua) pin digunakan untuk *inputan* buka dan tutup *gate* manual, 1 (satu) pin untuk sensor *gate*. Sedangkan pin *ouput* yang dipakai sebanyak 6 (enam) pin. Satu digunakan untuk *alarm / buzzer*, 1 (satu) pin sebagai lampu jalan, 4 (empat) pin untuk motor pembuka dan penutup *gate*. Berikut gambar *wiringnya* :



Gambar 3.3. Blok *wiring*

Tabel 3.1 Alokasi pin pada PLC

I/O	PLC 00	PLC 01
Input	Pin 00 (IR00000) Sensor <i>gate</i> pintu masuk	Pin 00 (IR00000) Sensor <i>gate</i> pintu keluar
	Pin 01–04 (IR00001-IR00004) Tombol <i>Emergency</i>	Pin 01-04 (IR00001-IR00004) Tombol <i>Emergency</i>
	Pin 05 (IR00005) Buka <i>gate</i> masuk manual	Pin 05 (IR00005) Buka <i>gate</i> keluar manual
	Pin 06 (IR00006) Tutup <i>gate</i> masuk manual	Pin 06 (IR00006) Tutup <i>gate</i> keluar manual

inp
com

0

1

2

3

4

Tabel 3.1 Alokasi pin pada PLC (Lanjutan)

I/O	PLC 00	PLC 01
Input	Pin 07 (IR00007) Saklar lampu disisi kiri jalan	Pin 07 (IR00007) Saklar lampu disisi kanan jalan
Output	Pin 00 (IR01000) <i>Buzzer</i>	Pin 00 (IR01000) <i>Buzzer</i>
	Pin 01 (IR01001) Lampu jalan disisi kiri	Pin 01 (IR01001) Lampu jalan disisi kanan
	Pin 04–07 (IR01004–IR01007) Kontrol motor <i>gate</i> pintu masuk	Pin 04–07 (IR01004–IR01007) Kontrol motor <i>gate</i> pintu keluar

3.3. Rangkaian Hardware

Dalam tugas akhir ini digunakan 2 buah PLC dengan sebuah PC sebagai *Master* (pengendali). Dimana kedua PLC tersebut difungsikan sebagai *slave* (dikendalikan) yang dihubungkan dengan *RS 232 to RS 485 Converter* untuk mengubah sistem komunikasi dari RS 232 ke sistem komunikasi RS 485. Topologi sistem yang digunakan adalah topologi *multipoint bus (multidrop)*. Kabel yang digunakan adalah 2 *twisted pair*, untuk *transmitter* dan *receiver* dan juga sebuah kabel untuk *ground*. Dalam sistem *Four Wire Multidrop Network*, *bus driver (transmitter)* dari *master* dihubungkan ke seluruh *receiver* dari *slave* dan seluruh *transmitter* dari *slave* akan masuk ke *receiver* dari *master*. Hubungan dari *master* ke *slave* menggunakan hubungan yang bersilangan (*transmitter* dari *master* masuk ke *receiver slave*), sedangkan hubungan antara *slave* menggunakan hubungan yang lurus atau sejajar (*transmitter* dari *slave* masuk ke *transmitter slave* yang lain).

3.3.1. RS 232 to RS 485 Converter

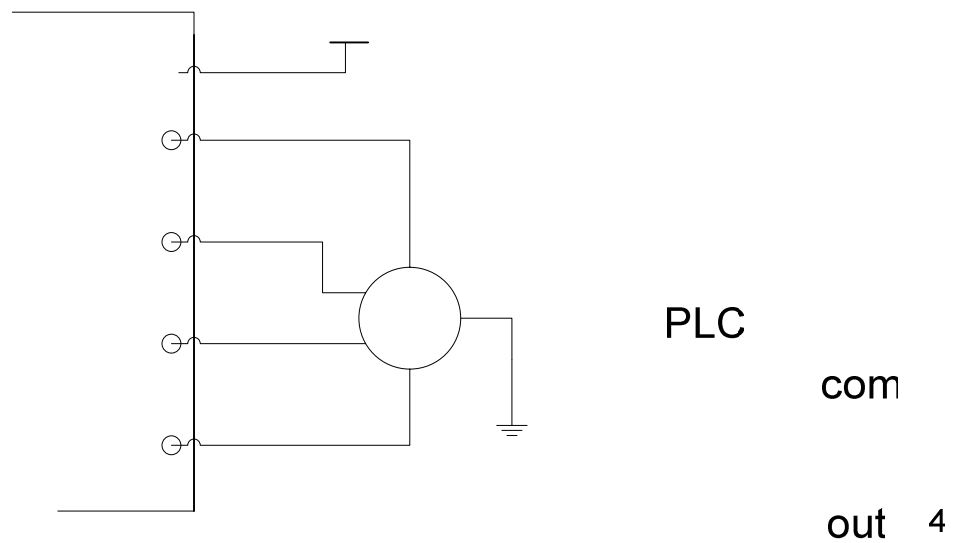
Pada dasarnya *RS232 to RS485 Converter* ini hanya merubah *level* tegangan saja. Apabila ada sinyal yang masuk dari PC ke IC MAX232, maka IC ini akan merubah *level* tegangan yang masuk menjadi *level* tegangan TTL. Dalam IC MAX 491 terdapat *Receiver Enable* (aktif low) dan *Driver Enable* (aktif high) untuk mengatur lalu lintas data dari level tegangan TTL ke level tegangan Differential. *Receiver Enable* (RE) berfungsi untuk memposisikan IC MAX 491 sebagai penerima, sedangkan *Driver Enable* memposisikan IC MAX 491 sebagai pengirim atau bisa juga disebut DE merupakan *control* dari IC MAX 491. Sistem

RS485 mempunyai DE sebagai *control* yang dapat memutuskan hubungan dari jalur transmisi ketika tidak mengirimkan data. *Converter* ini menggunakan RTS untuk mengontrol sinyal dari serial *port asynchronous* ke DE dari IC MAX 491. Jalur RTS yang terhubung dengan MAX 491 saat RTS berlogic 1 akan mengaktifkan RS 485. Jika RTS berlogic 0 maka MAX 491 dalam kondisi *tristate*. Pengaruh dari kondisi ini adalah memutuskan *driver* dari jalurnya.

Setelah IC MAX491 menerima *bit-bit* dari IC MAX232, maka akan diubah menjadi perbedaan tegangan antara dua terminal Y dan Z (terminal pada *transmitter*) sesuai dengan *logic bit* yang diterimanya. Begitu pula pada saat IC MAX491 menerima data dari MAX491 lain, IC ini akan mengubah perbedaan antara tegangan *terminal A* dan B (*terminal* pada *receiver*) menjadi tegangan TTL. Karena *receiver enable* (RE) dari IC MAX491 dihubungkan ke *ground*, maka *receiver* akan selalu aktif dan akan menerima langsung *logic* yang diterima dan diteruskan ke IC MAX232, yang selanjutnya oleh IC MAX232 ini akan diubah menjadi *level* tegangan RS-232 sesuai dengan *logic* yang diterimanya dan dimasukkan ke dalam *serial* pada PC. Untuk lebih jelasnya dapat dibaca di buku TA milik Sdr. Edison Lao (Tugas Akhir : No. 02010837/ELK/2007).

3.3.2. Rangkaian Kontrol *Gate*

Gate didesain untuk dapat memutar dua arah, yaitu searah jarum jam untuk membuka pintu *gate* dan berlawanan arah jarum jam untuk menutup pintu *gate*. Pada motor ini memakai supply 5V yang langsung dihubungkan ke PLC dengan menggunakan *sourcing mode*.



Gambar 3.4. Rangkaian Motor Stepper

Pada tugas akhir ini digunakan motor stepper 2 phase unipolar PM *type* stepper motor dengan 48-poles.



Gambar 3.5. Motor Stepper 42SPM-24SCG

Step angle : 7.5 degrees (360 degrees/48 poles)

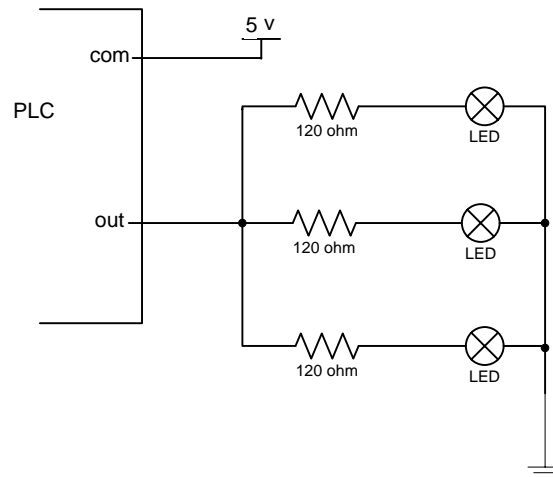
Drive voltage : 5 to 6V

Coil current : 250mA(5V)/2 coils

Untuk mendeteksi apakah *gate* sudah menutup atau masih membuka menggunakan *proximity switch sensor* sebagai *input* PLC. Apabila *gate* menutup maka akan terdeteksi oleh *proximity switch sensor* tersebut. Sehingga PLC dapat mengetahui apakah *gate* benar-benar sudah menutup.

3.3.3. Rangkaian Lampu Jalan

Pada simulasi disini menggunakan lampu dari LED yang kira-kira membutuhkan tegangan 2,5 V sehingga dibutuhkan tahanan tambahan supaya LED tidak rusak. Apabila dalam penerapan dalam dunia nyata yang menggunakan lampu jalan biasa akan dibutuhkan beberapa penyesuaian pada rangkaiannya.

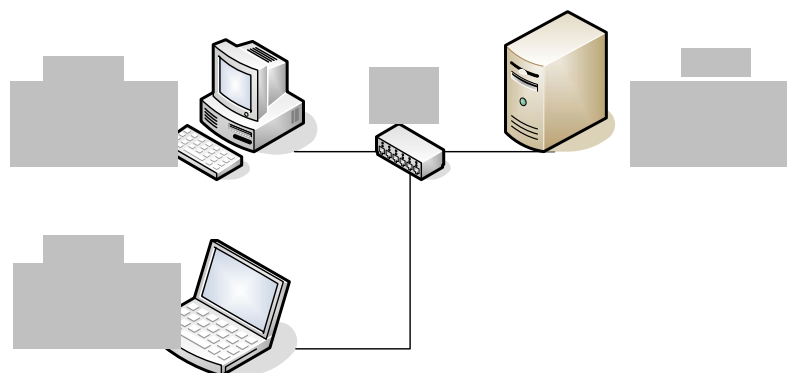


Gambar 3.6. Rangkaian Lampu Jalan

3.4. Konfigurasi Jaringan

Jaringan komputer yang digunakan adalah suatu jaringan *server-client* yang berskala *Local Area Network (LAN)* dengan menggunakan sebuah *switch hub*. *Server* pada tugas akhir ini sebagai penyedia layanan *database*, aplikasi, dan pengakses menuju PLC.

Berikut ini adalah konfigurasi jaringan komputer :



Gambar 3.7. Jaringan komputer

Semua aplikasi dan data-data yang dibutuhkan dimasukkan dalam *server*, komputer *client* hanya tinggal mengakses data-data tersebut serta aplikasinya. Sebelumnya komputer *client* juga harus diinstal program Wonderware InTouch 9.0 yang digunakan untuk menjalankan aplikasi tersebut.

3.5. Desain Database

Dalam penggunaan *database*, Wonderware InTouch dapat terintegrasi dengan bermacam-macam program *database server* salah satunya adalah Microsoft Access yang akan digunakan pada tugas akhir ini.

Untuk membuat *database* menggunakan Microsoft Access, desain sebuah tabel kemudian isi *Field name* dan *Data type*-nya, sebagai keterangan juga dianjurkan untuk menulis *description*-nya. Untuk *database* operator dapat dibuat seperti pada tabel berikut ini.

Tabel 3.2. Tabel operator

<i>Field name</i>	<i>Data type</i>	<i>Description</i>
No	<i>Auto number</i>	Sebagai <i>primary key</i>
NamaOperator	<i>Text</i>	Menampung nama operator
JamMasuk	<i>Date/Time</i>	Jam ketika <i>login</i>
JamKeluar	<i>Date/Time</i>	Jam ketika <i>logout</i>
Lokasi	<i>Text</i>	Lokasi tempat operator <i>login</i>
TanggalMasuk	<i>Date/Time</i>	Tanggal ketika operator <i>login</i>
tanggalKeluar	<i>Date/Time</i>	Tanggal ketika operator <i>login</i>

Setelah membuat desain tabel, set *primary key* dari tabel tersebut. Untuk *primary key* data yang ada di dalamnya harus unik dan tidak ada data yang kembar.

Sedangkan untuk tabel *alarm*, proses pembuatannya sama seperti di atas.

Tabel 3.3. Tabel *alarm*

<i>Field name</i>	<i>Data type</i>	<i>Description</i>
No	<i>Auto number</i>	Sebagai <i>primary key</i>
Tempat	<i>Text</i>	Tempat dimana <i>alarm</i> berbunyi
Jam	<i>Date/Time</i>	Jam ketika <i>alarm</i> berbunyi
Tanggal	<i>Date/Time</i>	Tanggal ketika <i>alarm</i> berbunyi
OperatorA	<i>Text</i>	Operator yang sedang berada di pos A ketika <i>alarm</i> berbunyi

Tabel 3.3. Tabel *alarm* (Lanjutan)

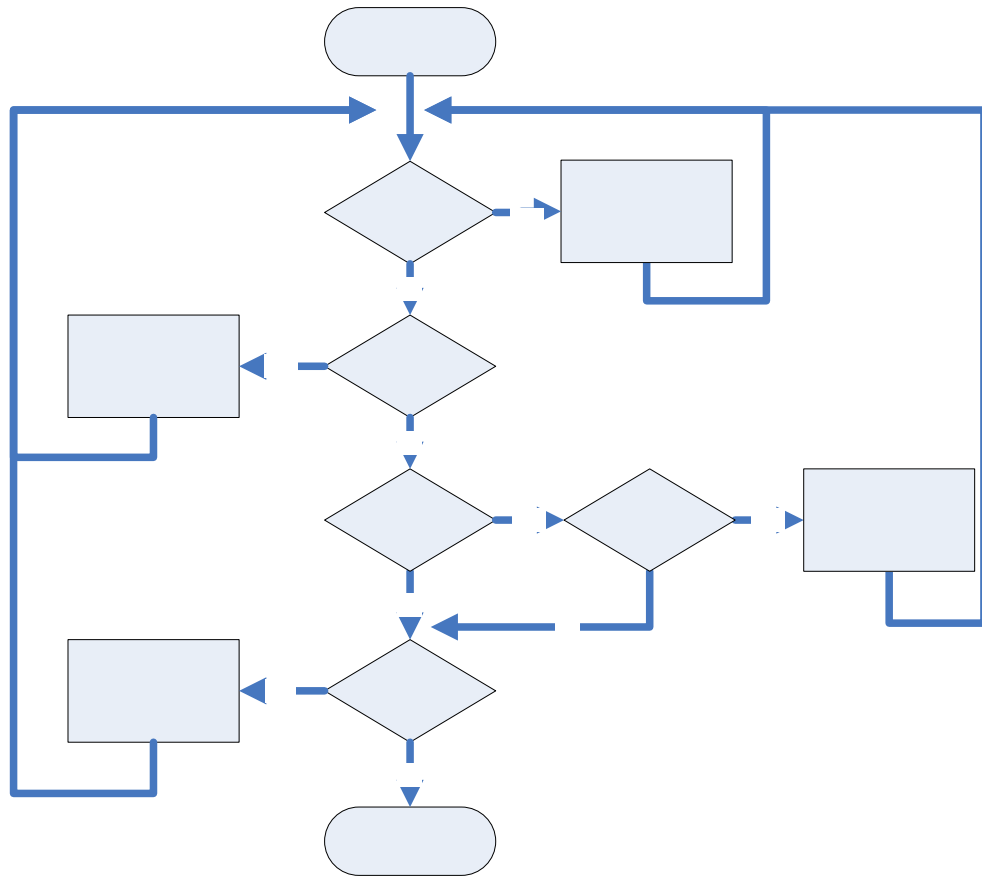
<i>Field name</i>	<i>Data type</i>	<i>Description</i>
OperatorB	<i>Text</i>	Operator yang sedang berada di pos B ketika <i>alarm</i> berbunyi
OperatorC	<i>Text</i>	Operator yang sedang berada di <i>server</i> ketika <i>alarm</i> berbunyi

3.6. Perancangan *Software*

Dalam tugas akhir ini menggunakan beberapa program tambahan selain Wonderware InTouch 9.0, program-program tambahan tersebut digunakan untuk men-*setting* PLC, membuat *ladder diagram*, dan serta komunikasi antar komputer dan PLC. Selain program Wonderware InTouch 9.0 diperlukan *ladder diagram* untuk mengontrol *ouput* dari PLC

3.6.1. *Ladder* PLC

Dalam pembuatan *ladder diagram* menggunakan software dari OMRON yaitu SYSWIN 3.3. *Ladder* ini berfungsi untuk kontrol sistem pada PLC dengan *flowchart* sebagai berikut :



Gambar 3.8. Flowchart pemrograman pada PLC. Untuk ladder diagram yang digunakan dapat dilihat pada lampiran.

3.6.2. Human Machine Interface (HMI) Wonderware InTouch 9.0

Program yang digunakan untuk mengontrol dan memonitoring plant adalah Wonderware InTouch 9.0. Program ini tidak hanya untuk tampilan Human Machine Interface-nya saja melainkan digunakan juga untuk penerapan SCADA menggunakan sistem Distributed Application.

Dalam HMI Wonderware InTouch sistem SCADA, untuk menginisialisasi input dan output dari PLC digunakan suatu tagname. Tagname-tagname tersebut digunakan sebagai media yang mendukung untuk mengambil dan mengirim data dari dan ke PLC, membuat animasi grafis untuk menggambar proses yang terjadi, perhitungan dan media kontrol untuk mengontrol PLC.

Start

Tombol Emergency ditekan?

No

Tekan Switch buka Gate?

No

Tekan Switch tutu Gate?

No

Tombol Lampu ditekan?

No

Tabel 3.4. *Tagname*

TAGNAME	TIPE	FUNGSI
\$DateString	System Message	Animation Link Use Bindlist Use
\$Hour	System Integer	Application Script Use
\$OperatorEntered	System Message	Animation Link Use Bindlist Use
\$Second	System Integer	Data Change Script Use
\$TimeString	System Message	Animation Link Use Bindlist Use
AlarmRumah	GroupVar	Animation Link Use
BindList	Memory Message	Animation Link Use Data Change Script Use
BLINK	I/O discrete	Animation Link Use
BukaKunci	I/O discrete	Animation Link Use Application Script Use
Counter	Memory integer	Animation Link Use Data Change Script Use
ErrorMsg	Memory Message	Animation Link Use Data Change Script Use
Heartbeat	I/O integer	Data Change Script Use
HistTrend1	Hist Trend	Animation Link Use
HR00001	I/O discrete	Animation Link Use Condition Script Use
HR00002	I/O discrete	Animation Link Use Condition Script Use
HR00003	I/O discrete	Animation Link Use Condition Script Use
HR00004	I/O discrete	Animation Link Use Condition Script Use
HR0001	I/O discrete	Animation Link Use Condition Script Use
HR0002	I/O discrete	Animation Link Use Condition Script Use
HR0003	I/O discrete	Animation Link Use Condition Script Use
HR0004	I/O discrete	Animation Link Use Condition Script Use
HR00100	I/O discrete	Animation Link Use
HR00101	I/O discrete	Animation Link Use
HR0100	I/O discrete	Animation Link Use
HR0101	I/O discrete	Animation Link Use
IR0000	I/O discrete	Animation Link Use
IR00000	I/O discrete	Animation Link Use
IR01001	I/O discrete	Animation Link Use

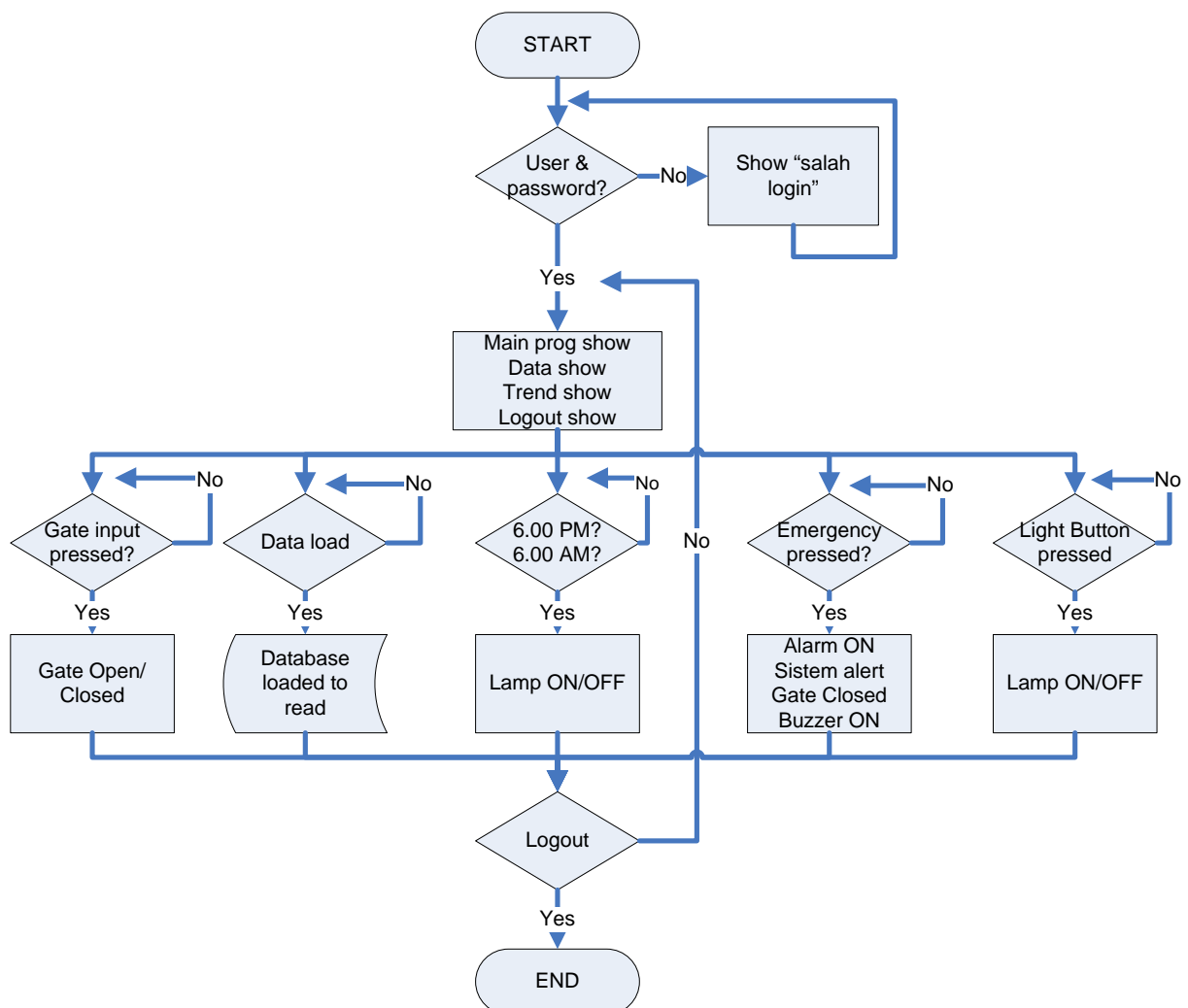
Tabel 3.4. *Tagname* (Lanjutan)

TAGNAME	TIPE	FUNGSI
IR01100	I/O discrete	Animation Link Use
IR1001	I/O discrete	Animation Link Use
IR1100	I/O discrete	Animation Link Use
JamKeluar	Memory Message	Animation Link Use
		Bindlist Use
JamMasuk	Memory Message	Animation Link Use
		Bindlist Use
JumlahAlarm	Memory Real	Bindlist Use
		Condition Script Use
		Data Change Script Use
		Historical Trend Pen Use in
JumlahGateKeluar	Memory Real	Historical Trend Pen Use in
JumlahGateMasuk	Memory Real	Historical Trend Pen Use in
ON	Memory Message	Animation Link Use
		Condition Script Use
OperatorA	I/O Message	Animation Link Use
		Bindlist Use
OperatorB	I/O Message	Animation Link Use
		Bindlist Use
OperatorC	I/O Message	Animation Link Use
		Bindlist Use
Remote:HR00001.Ack		Animation Link Use
Remote:HR00002.Ack		Animation Link Use
Remote:HR00003.Ack		Animation Link Use
Remote:HR00004.Ack		Animation Link Use
Remote:HR0001.Ack		Animation Link Use
Remote:HR0002.Ack		Animation Link Use
Remote:HR0003.Ack		Animation Link Use
Remote:HR0004.Ack		Animation Link Use
reset	Memory Discrete	Animation Link Use
		Condition Script Use
TableName	Memory Message	Animation Link Use
		Data Change Script Use
TanggalKeluar	Memory Message	Animation Link Use
		Bindlist Use
TanggalMasuk	Memory Message	Animation Link Use
		Bindlist Use
Tempat	Memory Message	Bindlist Use
		Condition Script Use
Template	Memory Message	Animation Link Use
TemplateName	Memory Message	Animation Link Use

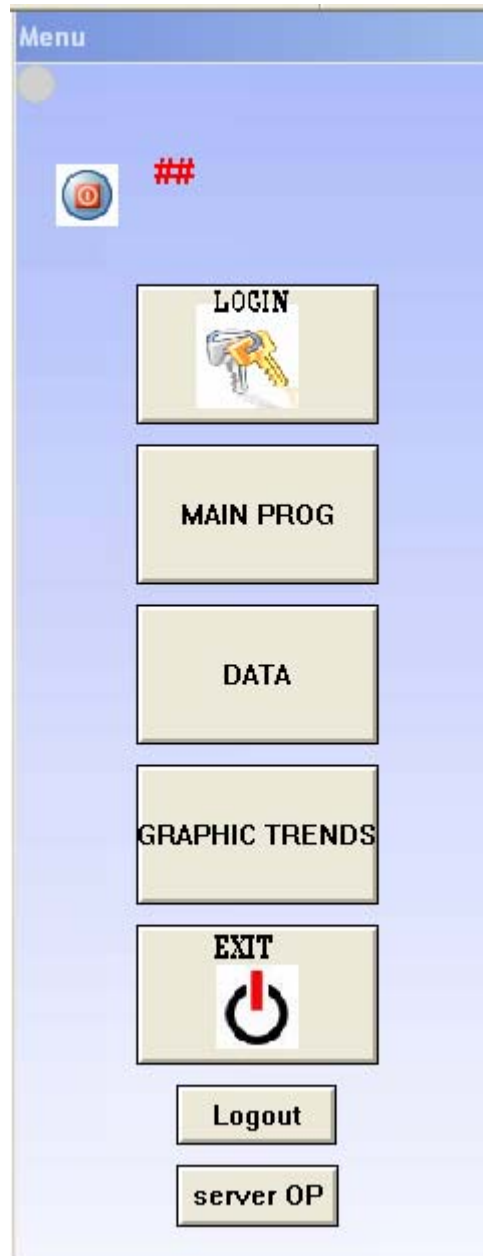
Tabel 3.4. *Tagname* (Lanjutan)

TAGNAME	TIPE	FUNGSI
Username	Memory Message	Animation Link Use
		Bindlist Use
UserPassword	Memory Message	Animation Link Use
		Bindlist Use
Waktu	Memory Message	Animation Link Use
		Bindlist Use

Berikut ini *flowchart* dari sistem untuk penggunaan secara otomatis menggunakan HMI :

Gambar 3.9. *Flowchart* Sistem menggunakan HMI

HMI pada tugas akhir ini menggunakan beberapa *windows* sebagai tampilan dari sistem dan fitur-fitur dari Wonderware InTouch. Pada tampilan awal terdapat *window* Menu, serta beberapa pilihan tombol yang akan mengarahkan kepada *windows* yang akan dituju.



Gambar 3.10. Menu HMI



Gambar 3.11. Tampilan *window* Operator.

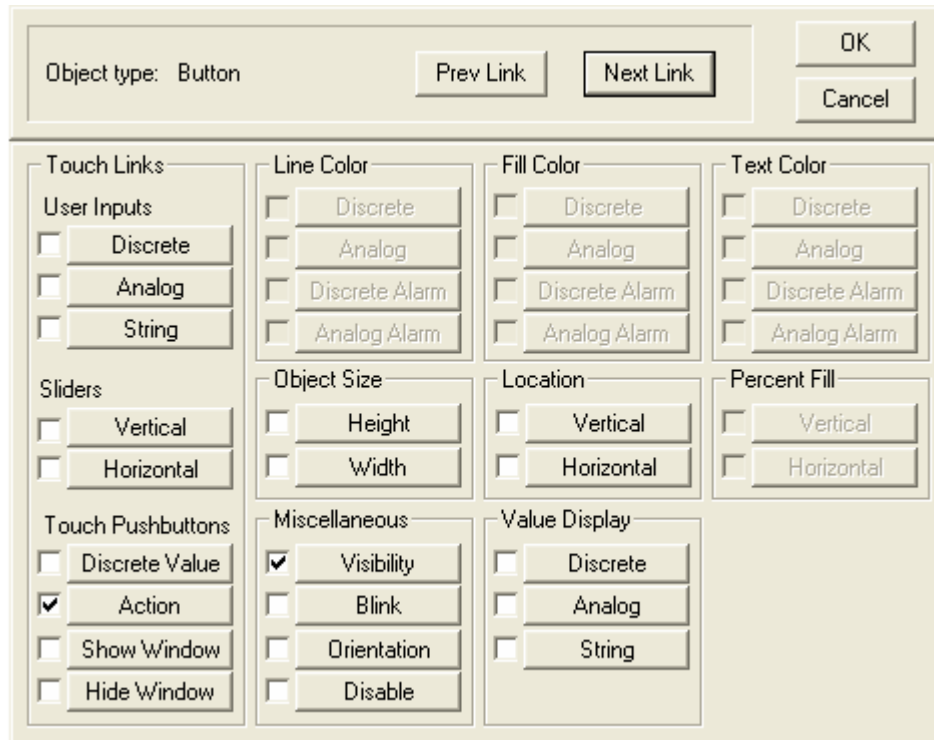
Untuk menghindari *client* yang berada di lokasi A melakukan *login* pada posisi yang tidak seharusnya (demikian juga untuk lokasi B), aplikasi melakukan identifikasi *node* pada komputer di pos tersebut sehingga hanya bisa *Login* di posisi yang telah ditentukan oleh server.

Pada *script* diberi perintah :

```
GetNodeName>NamaNode, 15 );
```

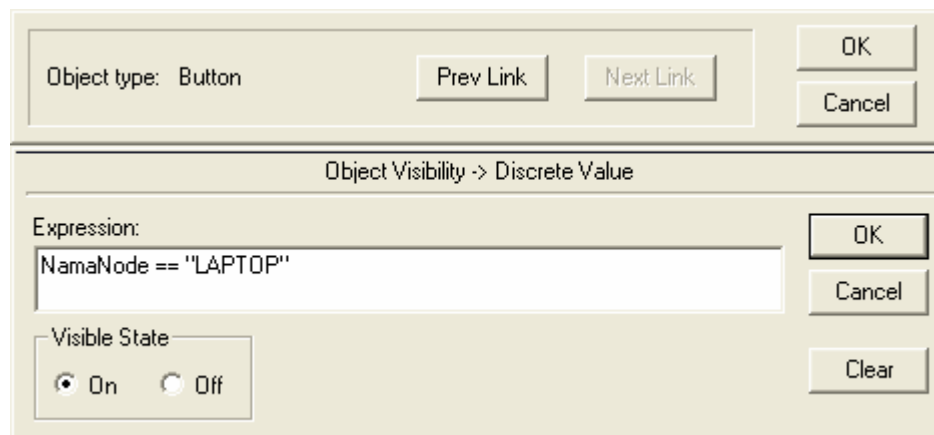
Pada perintah tersebut *nodeName* dari komputer yang mengakses aplikasi tersebut akan disimpan pada *TagName* "NamaNode", dan nama komputer tersebut maksimal 15 karakter.

Kemudian pada tombol "POS A" dilakukan konfigurasi *visibility* pada *Animation Links*.



Gambar 3.12. *Animation Links*

Pada *Object Visibility* diberi perintah apakah NamaNode == “Nama Komputer yang diperuntukkan khusus POS A” seperti pada gambar berikut.



Gambar 3.13. *Object Visibility*

Begitu juga dilakukan untuk POS B.

Sedangkan untuk konfigurasi *username* dan *password* dapat dibuat pada menu *Action* pada *Animation Link* dan dimasukkan *script* sebagai berikut.

Script untuk *login* operator untuk tombol pos A:

```

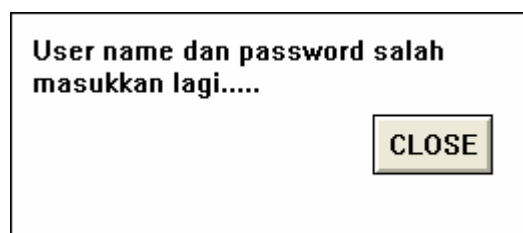
SQLDisconnect( ConnectionId );
Username = $OperatorEntered;
OperatorA = $OperatorEntered;

IF (UserPassword == "petra") AND ($OperatorEntered == "elektro") THEN
BukaKunci = 1 ;
TanggalMasuk = $DateString;
Lokasi = "A";
JamMasuk = $TimeString;
UserPassword = "";

ELSE
  IF (UserPassword == "123") AND ($OperatorEntered == "maspung") THEN
    TanggalMasuk = $DateString;
    Lokasi = "A";
    JamMasuk = $TimeString;
    BukaKunci = 1;
    UserPassword = "";
  ELSE
    IF (UserPassword == "456") AND ($OperatorEntered == "okta") THEN
      TanggalMasuk = $DateString;
      Lokasi = "A";
      JamMasuk = $TimeString;
      BukaKunci = 1;
      UserPassword = "";
    ELSE
      IF (UserPassword == "789") AND ($OperatorEntered == "angga") THEN
        TanggalMasuk = $DateString;
        Lokasi = "A";
        JamMasuk = $TimeString;
        BukaKunci = 1;
        UserPassword = "";
      ELSE
        BukaKunci = 0;
        Show "salah login";
      ENDIF;
    ENDIF;
  ENDIF;
ENDIF;
Hide "Operator";

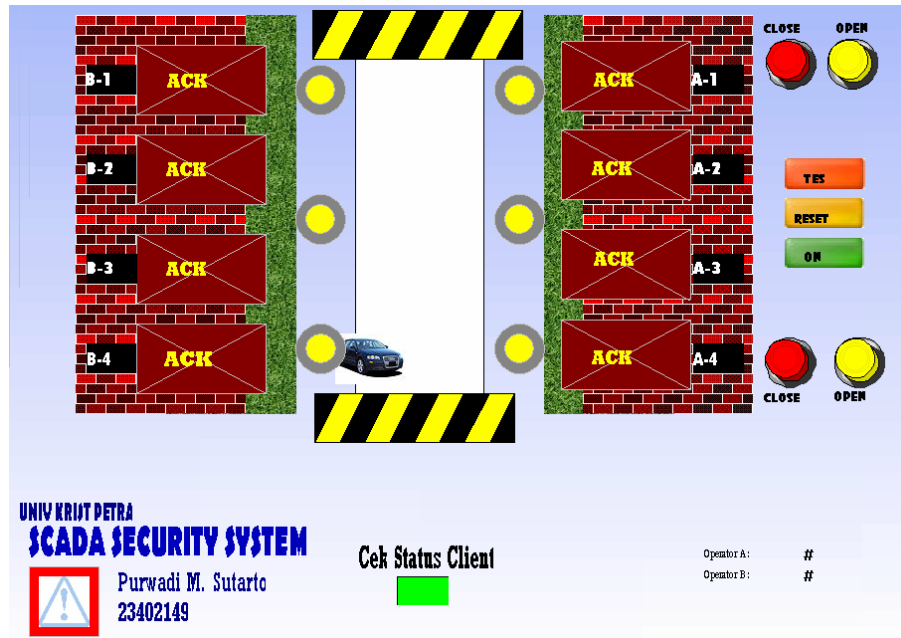
```

Seperti halnya tombol pos A, tombol pos B juga menggunakan *script* yang sama hanya diubah pada “lokasi = “B””.

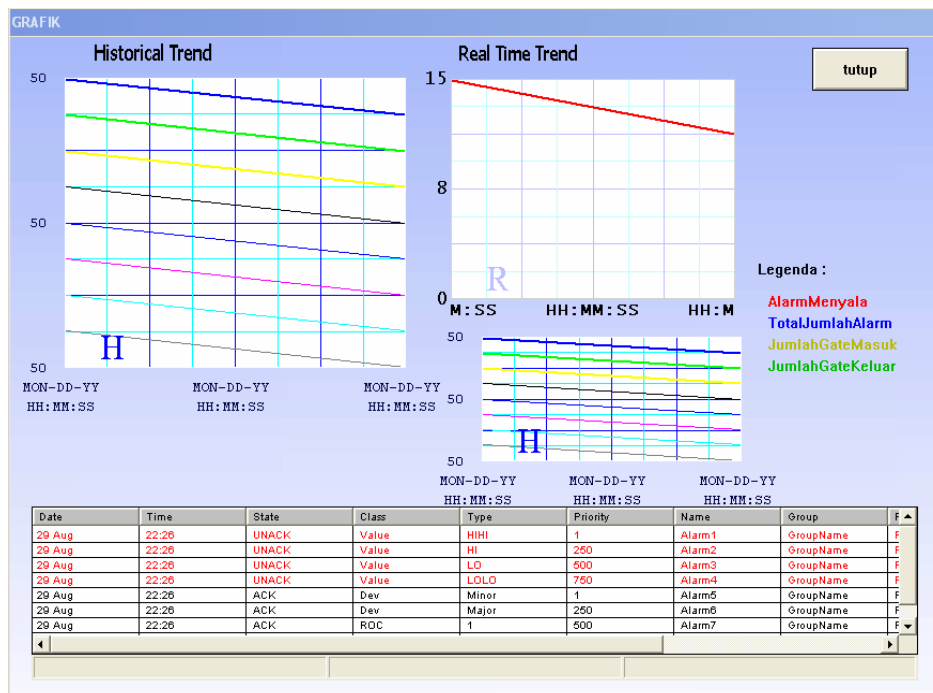


Gambar 3.14. Salah *login* pop up

Pada *main* window terdapat tampilan dari perumahan tersebut yang berfungsi untuk mendeteksi rumah mana yang *alarmnya* menyala.

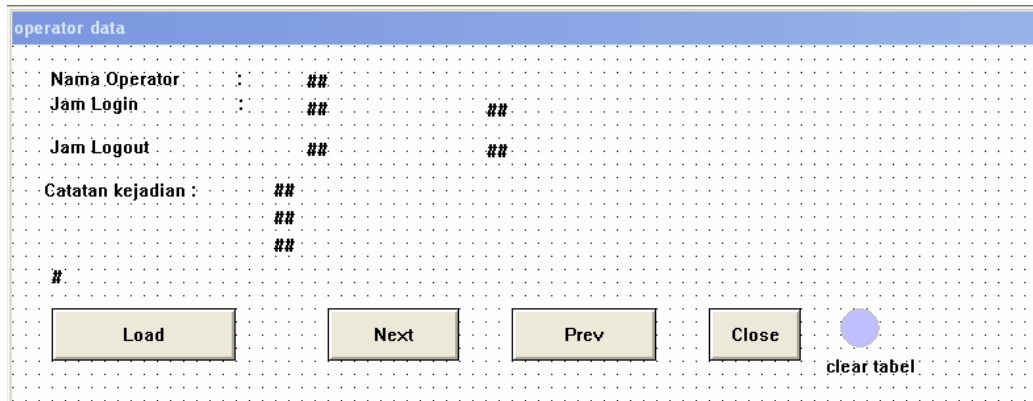


Gambar 3.15. Tampilan kompleks perumahan



Gambar 3.16. Tampilan *graphic trends*

Pada menu *graphic trends* terdapat *real time trend*, *historical trends* dan juga tabel *historical alarm*. Sedangkan tabel *real time alarm*, akan secara otomatis keluar apabila ada *alarm* yang menyala.



Gambar 3.17. Operator data

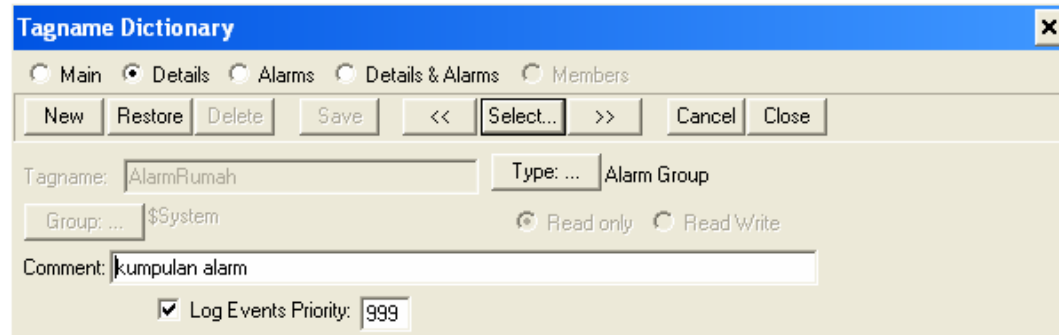
Sedangkan untuk menu data terdapat tampilan untuk melihat data-data yang telah ada dari *database* untuk jam dan tanggal berapa operator melakukan *login* dan *logout*.

3.6.3. Fitur Wonderware InTouch 9.0

3.6.3.1. Konfigurasi *Historical Logging*

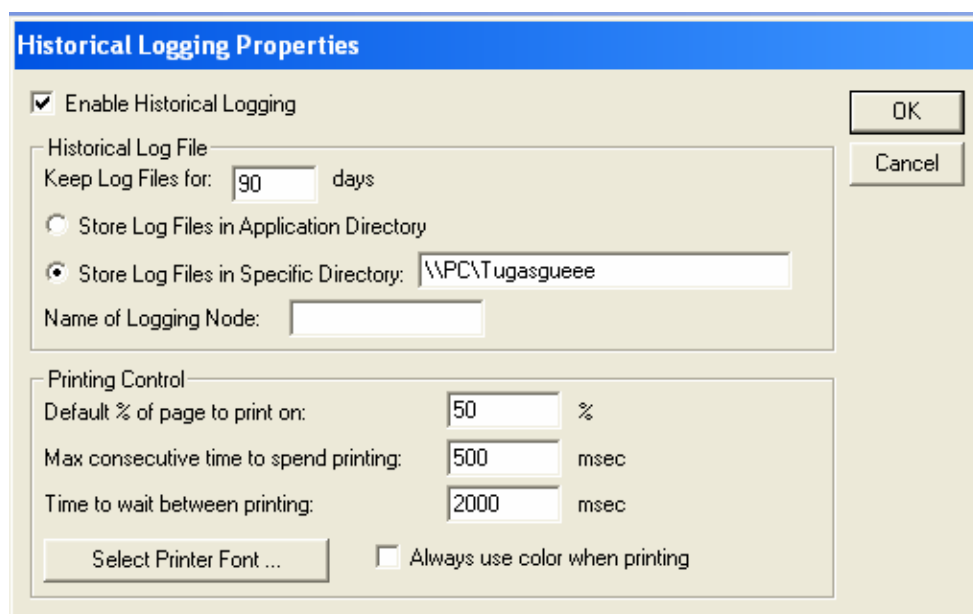
Pada Wonderware InTouch disediakan *Historical Logging* yang digunakan untuk menyimpan data-data pada *tagname* tertentu yang diinginkan. Termasuk akan perubahan data yang ada.

Untuk memilih *tagname* yang akan dimasukkan dalam *log*, masuk ke dalam *tagname dictionary* dan centang (✓) pada *log events*.



Gambar 3.18. Log pada tagname dictionary

Untuk konfigurasi dari *historical logging* tersebut masuk ke dalam menu *Special*, lalu ke sub menu *configure* dan klik pada *Historical logging*. Maka akan muncul tampilan seperti di bawah ini.



Gambar 3.19. Konfigurasi *historical logging*

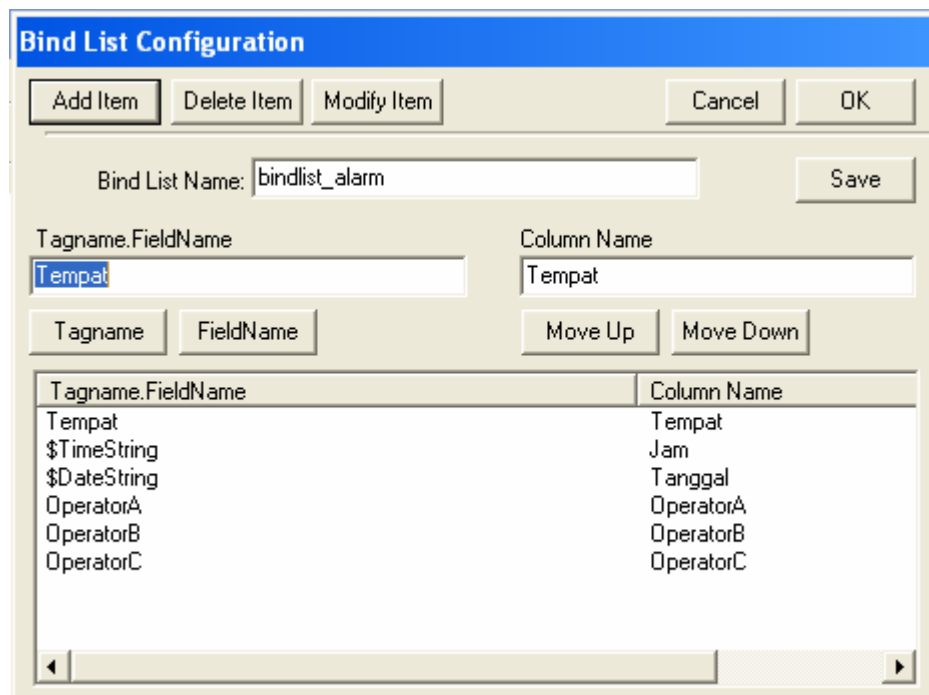
Untuk mengaktifkan *historical logging* centang (✓) pada *Enable Historical Logging* lalu set dimana penyimpanan data pada *Directory File*-nya.

3.6.3.2. SQL Access Manager pada Wonderware InTouch

Dalam Wonderware InTouch untuk mengakses data yang ada pada *database* digunakan *SQL Access Manager*. *SQL Access Manager* ini memberikan inisialisasi antara *tagname* pada wonderware dan kolom pada *database* selain itu

juga memberikan perintah kapan harus *write* ataupun membaca data yang ada di *database*.

Pada tugas akhir ini menggunakan *database* berbasis Microsoft Access untuk data *operator login* dan untuk *alarm history*. Untuk itu harus diatur *Bindlist* serta *Table Template* untuk inialisasi *tagname* ke kolom tabel *database* yang digunakan. Tabel yang digunakan sebanyak dua tabel yaitu Tabel_*Alarm* untuk data *alarm* dan Tabel_*2* untuk data operator.



Gambar 3.20. Konfigurasi “bindlist_alarm”

Gambar 3.21. Konfigurasi “bindlist_centre”

Table template digunakan untuk menyesuaikan *type* dari kolom tabel dengan yang ada pada *database*.

Column Name	Column Type	Length	Allow Nu
No	Number	50	Null
Tempat	Text	50	Null
Jam	Date/Time	50	Null
Tanggal	Date/Time	50	Null
OperatorA	Text	50	Null
OperatorB	Text	50	Null
OperatorC	Text	50	Null

Gambar 3.22. *Template* dari “Tabel_Alarm”

Column Name	Column Type	Length	Allow Null
NamaOperator	Text	131	Null
JamMasuk	Date/Time	50	Null
JamKeluar	Date/Time	50	Null
Lokasi	Text	131	Null
No	Number	50	Null
TanggalMasuk	Text	131	Null
TanggalKeluar	Text	131	Null

Gambar 3.23. *Template* dari “Tabel_2”

Untuk membaca ataupun memasukkan data antara *database* dan aplikasi InTouch menggunakan perintah berikut, yang ditulis pada *script* di InTouch :

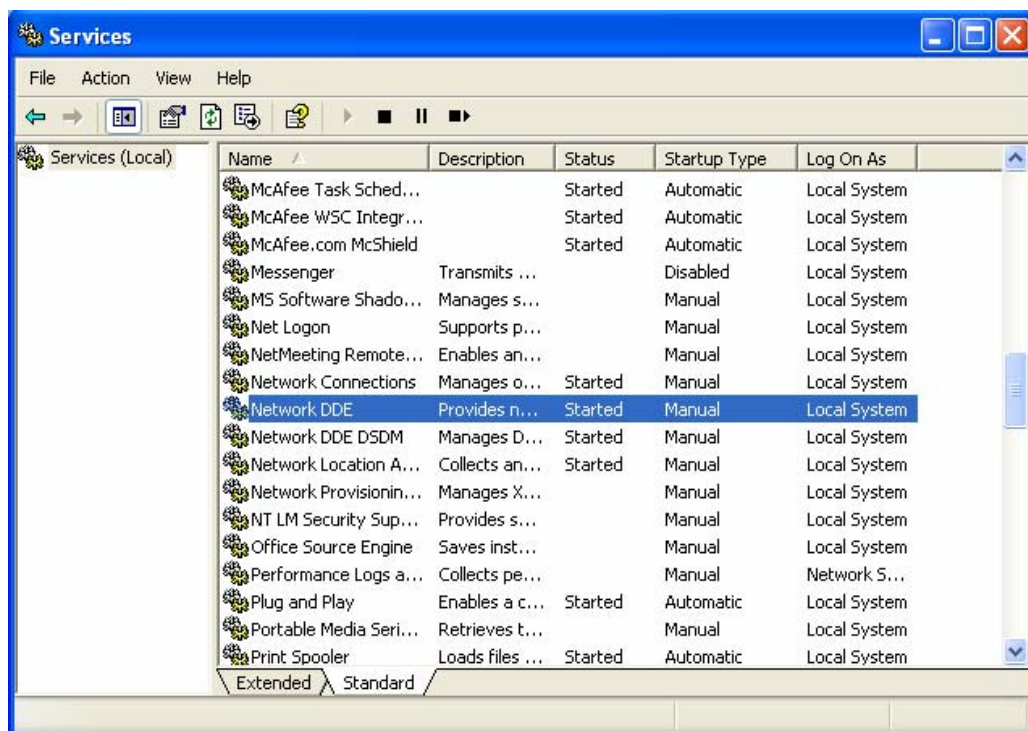
- Untuk *connect* ke *database* menggunakan Microsoft Access
SQLConnect(ConnectionId, "<attribute>=<value>; <attribute>=<value>;...");
 Contoh : **SQLConnect**(ConnectionId, “dsn=*database*db”);
- Untuk menghapus semua kolom pada tabel
SQLClearTable(ConnectionId, TableName);
- Untuk memberitahu apabila terjadi error melalui result code = -1
SQLErrorMsg(ResultCode);
- Untuk menuju nomor baris pada tabel
SQLNumRows(ConnectionId);
- Untuk mengambil data informasi dari tabel.
SQLSelect(ConnectionId, TableName, BindList, WhereExpr, OrderByExpr)
- Untuk memasukkan data InTouch tag ke satu baris tabel
SQLInsert(ConnectionId, TableName, BindList);
- Untuk disconnect dari *database*
SQLDisconnect(ConnectionId);

3.6.4. Konfigurasi *Distributed Application*

Aplikasi yang sudah dibuat dari komputer *server* termasuk *database* dan *file log* yang ada, disimpan dalam *folder* yang kemudian di-*share* dalam network sehingga dapat diakses oleh komputer *client*.

Sebelum mengakses aplikasi tersebut harus mengaktifkan protokol Network DDE dan Wonderware SuiteLink terlebih dahulu baik *server* maupun *client*.

Untuk mengaktifkan NetDDE dan SuiteLink pada Microsoft Windows dengan membuka “*services*” yang terdapat dalam *Administrative Tools* pada *Control Panel* di Windows XP, maka akan terlihat seperti pada tampilan berikut.

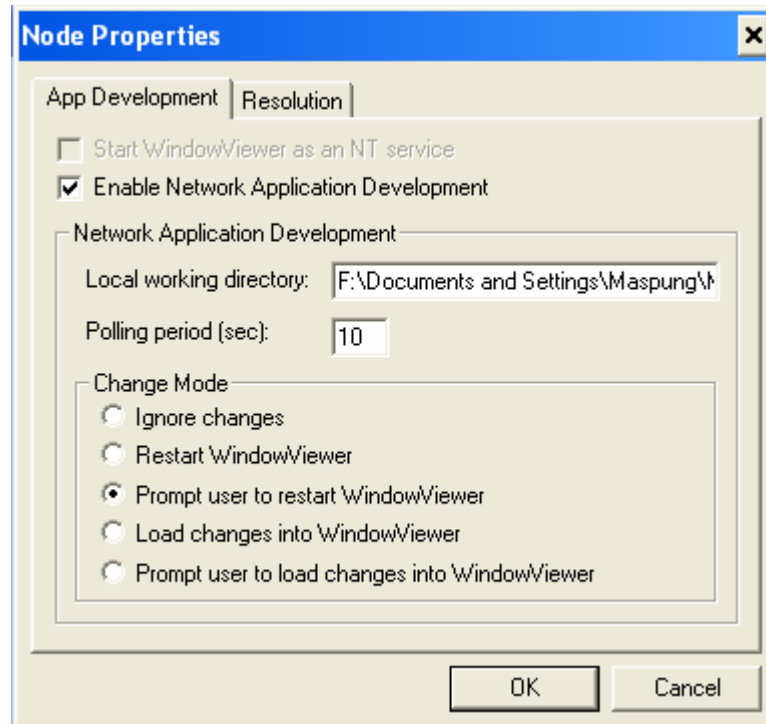


Gambar 3.24. Tampilan pada “*services*”

Kemudian status NetDDE dan Wonderware SuiteLink akan terlihat di daftar tersebut, untuk mengaktifkannya dapat di-start pada service tersebut sehingga merubah statusnya menjadi *started*.

Apabila komputer *client* ingin mengakses aplikasi tersebut, komputer *client* tinggal memasukkan *file application* yang sudah di-*share* oleh komputer *server* ke dalam InTouch Application Manager yang terdapat pada komputer *client*. Setelah memasukkan *application*, pada *Application Manager* juga

diperlukan untuk mengaktifkan *Network Application Development* dengan cara memberi tanda centang (✓) pada *Enable Network Application Development* yang terdapat di dalam *Node Properties*.



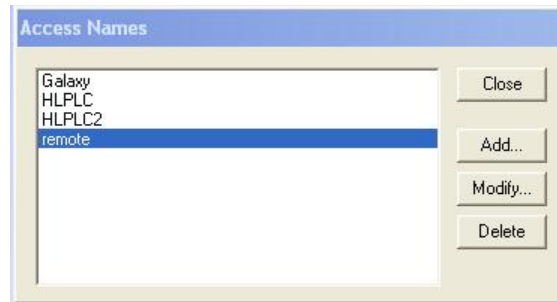
Gambar 3.25. *Node properties*

3.6.4.1. *Remote Tagname*

Pada SCADA diperlukan beberapa konfigurasi supaya hubungan antara *server* dan *client* dapat menjalankan aplikasi tersebut secara sinkron dan dapat terintegrasi. Selain *Distributed Application*, diperlukan juga *Remote tagname* untuk menyinkronkan *tagname server* dan *client*. Perubahan yang terjadi pada *tagname client* mengacu pada perubahan pada *Tagname* yang ada di *server* dan sebaliknya, sehingga *tagname* tersebut saling terkait.

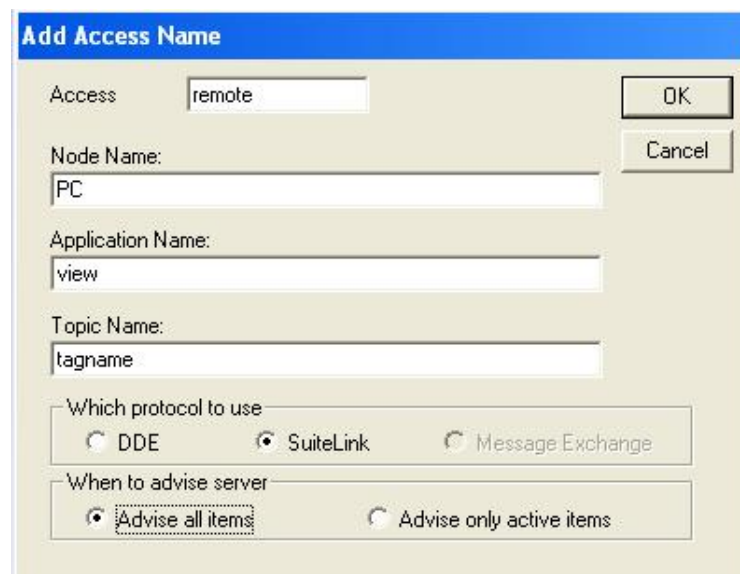
Berikut ini konfigurasi *Remote Tagname* :

Pada menu *Special* pilih *Access Names*.



Gambar 3.26. *Access Names window*

Kemudian tambah *Access Name* untuk *Remote Tagname* dengan memilih *Add* dan akan keluar tampilan seperti berikut.



Gambar 3.27. *Add access name window*

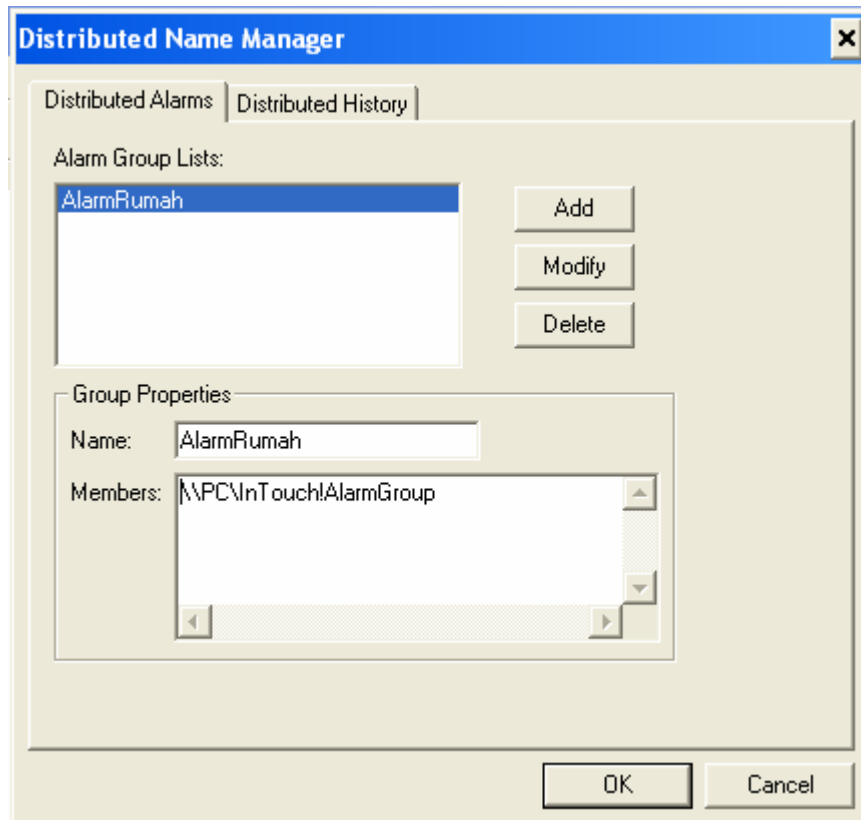
Pada kolom *Access* di beri nama “*Remote*”, kemudian *nodename* diisi nama *node* yang akan di akses *tagname*-nya. Kemudian *application name* diisi “*view*”, kemudian *Topic Name* diisi “*tagname*”. *Protocol* yang digunakan adalah *SuiteLink* dan *When to advise server* menggunakan *advise all items*.

Pada *Access Name* di atas digunakan pada aplikasi yang terdapat pada *client* untuk mengakses *tagname* yang berada pada *node PC (server)*.

Sedangkan untuk aplikasi yang terdapat pada *server* juga diperlukan *Remote Tagname* untuk mengakses *tagname* pada *client* yang ada yang digunakan untuk mengambil *tagname* nama operator yang *login* pada *client* tersebut. Konfigurasinya sama seperti di atas hanya saja nama *Access Name*-nya dirubah serta *Node Name* diisi nama komputer *client* tersebut.

3.6.4.2. Konfigurasi *Distributed Alarm* dan *Historical Alarm*

Salah satu fitur yang ada dalam Wonderware InTouch adalah *distributed alarm*, sehingga pada komputer lain (*client*) mendapatkan *alarm* yang sama dengan melalui *distributed alarm*. Konfigurasinya adalah sebagai berikut :



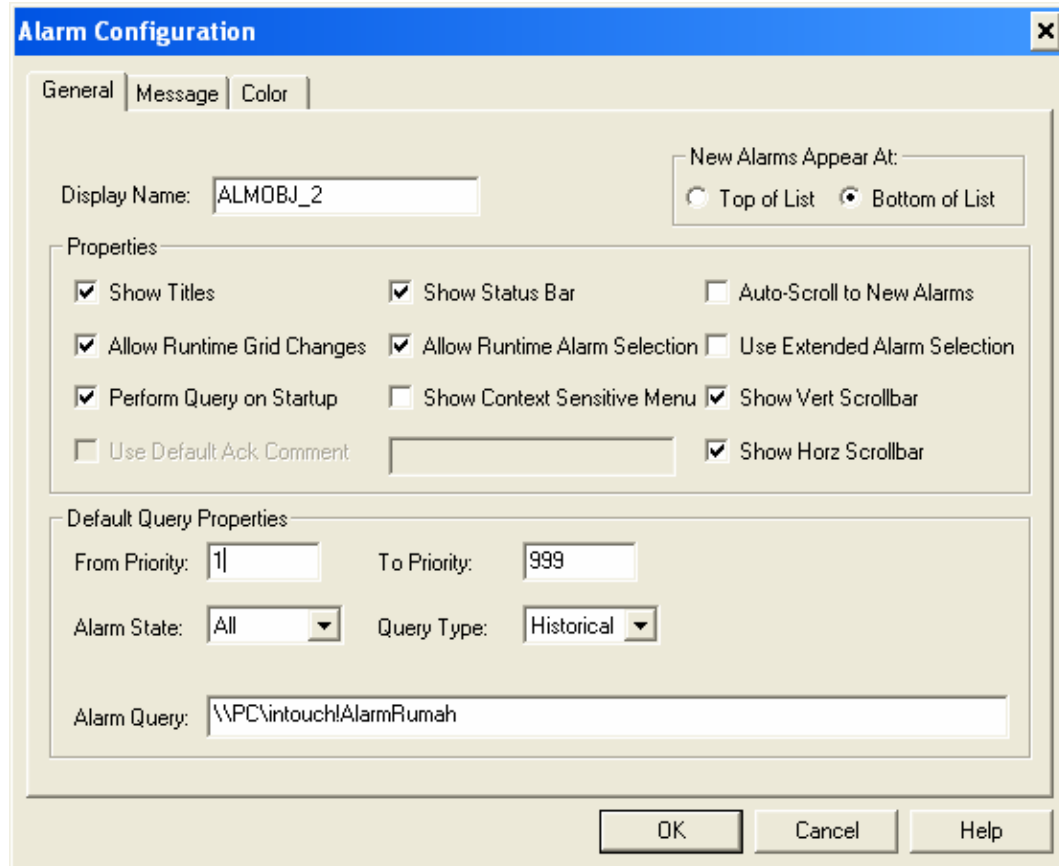
Gambar 3.28. *Distributed Alarm*

Pada Menu *Special*, lalu *configure*, kemudian buka link dari *Distributed Name Manager*. Dalam tampilan *distributed name manager* tersebut kita dapat memasukkan group *alarm* yang akan didistribusikan seperti pada contoh diatas.

Dalam suatu sistem yang terstruktur terdapat suatu *historical files* yang berfungsi untuk melihat riwayat dan keadaan yang terjadi sebelumnya. Dalam hal ini sangat diperlukan *history* dari *alarm*, berapa banyak *alarm* yang telah terjadi pada tanggal dan pukul berapa, dimana kejadiannya, dan sebagainya.

Untuk setelan konfigurasi pada *Historical alarm* maka buat tabel pada Wizard pilih *Distributed Alarm Display*. Kemudian pada *properties display* tersebut seperti pada gambar di bawah ini pada *box Query Type* diisi *Historical* dan pada *Alarm Query* diisi *Directory alarm* yang akan diakses, menggunakan

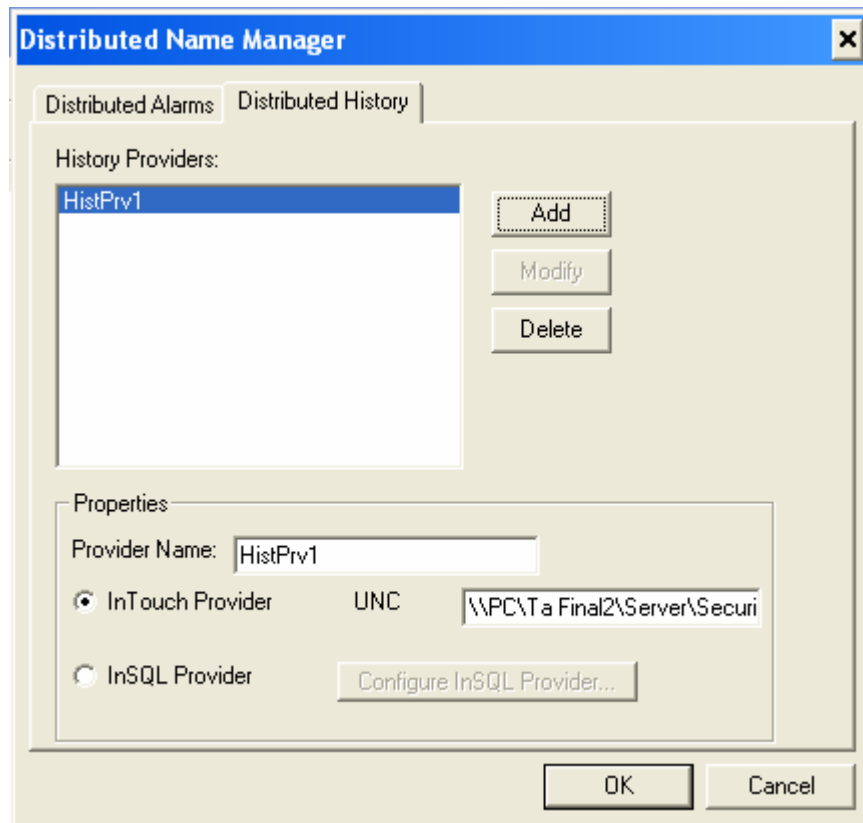
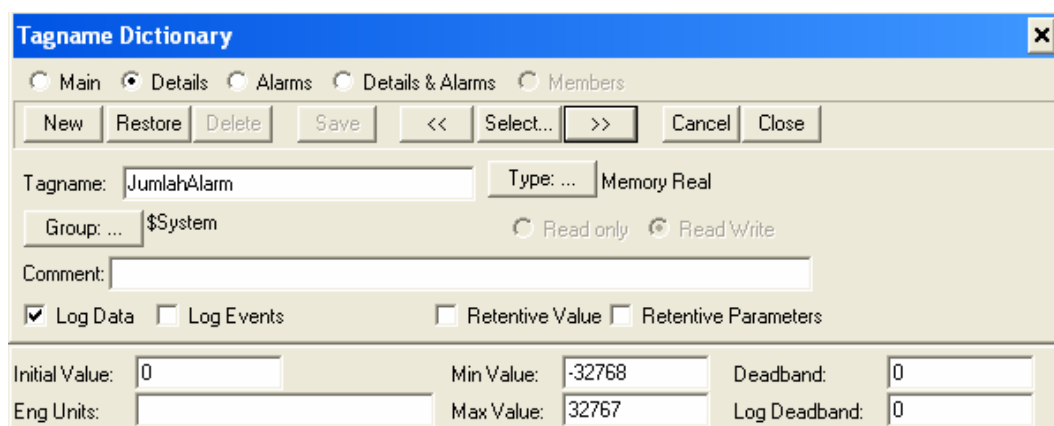
format “\\Node\intouch!AlarmGroup” untuk penerapannya dalam sistem tugas akhir ini menjadi “\\PC\intouch!AlarmRumah”.



Gambar 3.29. *Historical Alarm display configuration*

3.6.4.3. Konfigurasi *Distributed Historical Trends*

Dalam suatu *distributed historical trends* diperlukan beberapa langkah untuk membuatnya, pertama buatlah suatu provider list pada *Distributed Name Manager*, kedua buatlah atau konfigurasi suatu *historical trends object*, ketiga konfigurasi *distributed logging* pada aplikasi tersebut, keempat distribusikan aplikasi tersebut dengan *Network Application Development (NAD)*.

Gambar 3.30. *History provider*Gambar 3.31. *Historical trends object tagname*

Historical Trend Configuration

Historical Tag:

Chart time: Initial Time Span:
 Secs Mins Hrs Days

Initial Display Mode: Min/Max Average

Color: Chart Color:
 Border Color:

Time Divisions: Number of Major Div:
 Minor Div/Major Div:
 Top Labels Bottom Labels
 Major Div/Time Label:
 Time: MM DD YY
 HH MM SS

Value Divisions: Number of Major Div:
 Minor Div/Major Div:
 Left Labels Right Labels
 Major Div/Value Label:
 Min Value: Max:

Pen	Tagname	Color	Width	Pen	Tagname	Color	Width
1	<input type="text" value="HistPrv1.JumlahAlarm"/>	<input type="color" value="#0000FF"/>	<input type="text" value="2"/>	5	<input type="text"/>	<input type="color" value="#0000FF"/>	<input type="text" value="1"/>
2	<input type="text" value="HistPrv1.JumlahGateKeluar"/>	<input type="color" value="#00FF00"/>	<input type="text" value="2"/>	6	<input type="text"/>	<input type="color" value="#FF00FF"/>	<input type="text" value="1"/>
3	<input type="text" value="HistPrv1.JumlahGateMasuk"/>	<input type="color" value="#FFFF00"/>	<input type="text" value="2"/>	7	<input type="text"/>	<input type="color" value="#00FFFF"/>	<input type="text" value="1"/>
4	<input type="text"/>	<input type="color" value="#000000"/>	<input type="text" value="1"/>	8	<input type="text"/>	<input type="color" value="#808080"/>	<input type="text" value="1"/>

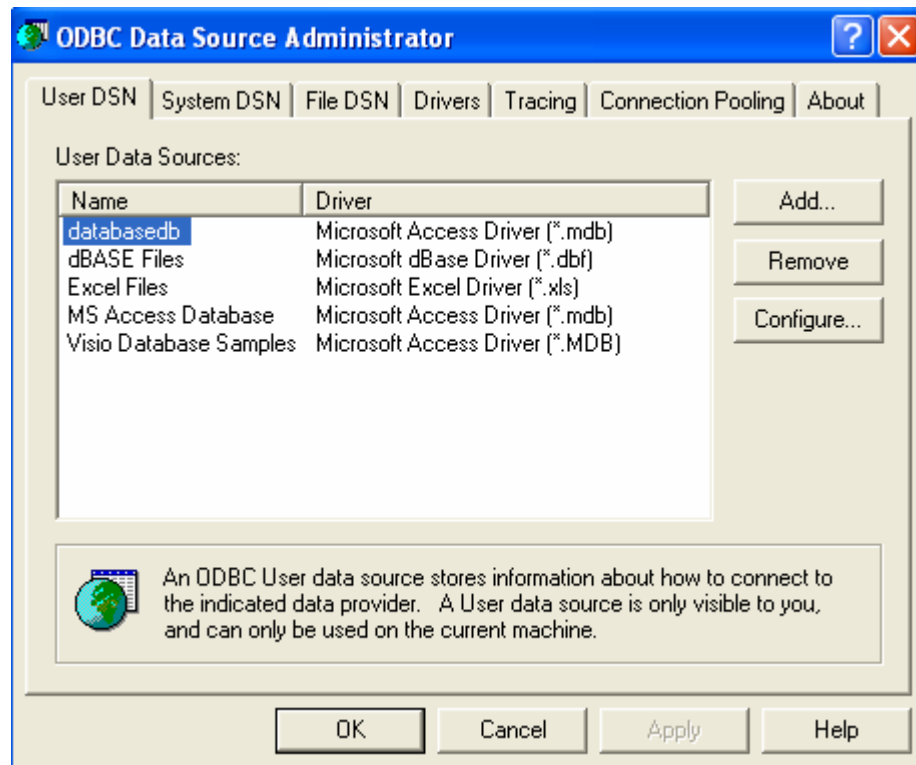
OK Cancel Clear Select Display Font ... Allow runtime changes

Gambar 3.32. *Historical trends configuration*

Pada tugas akhir ini menggunakan *distributed historical logging* pada *server* PC sebagai sumber data dari *historical trends*.

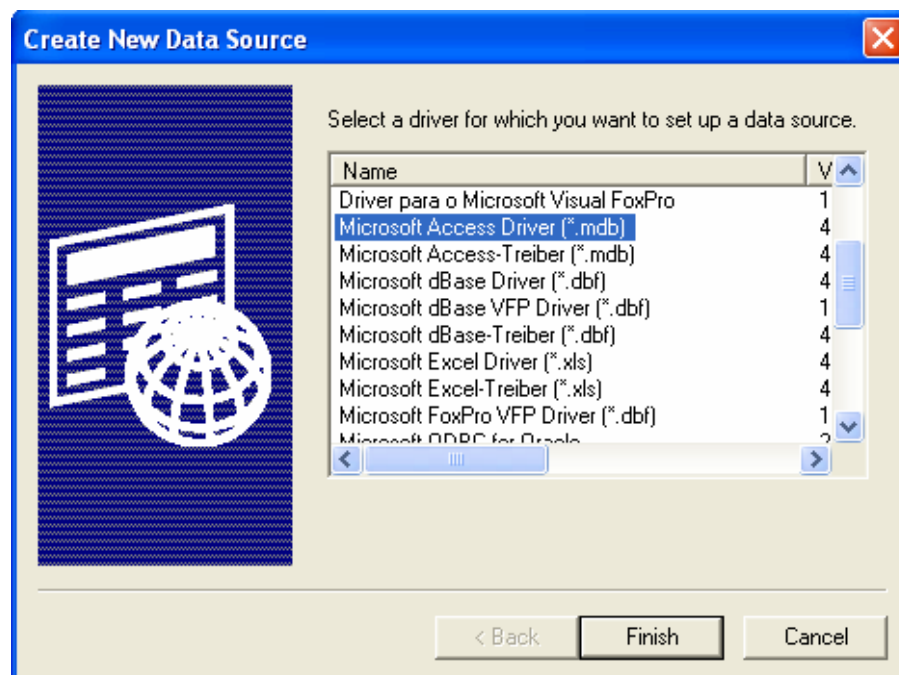
3.6.5. Konfigurasi *Database Connectivity* Pada Microsoft Windows.

Pada Microsoft Windows untuk mengakses *database* yang terdapat dalam *server* diperlukan konfigurasi *Open Database Connectivity* (ODBC) yang terdapat dalam *Administrative Tools* pada *Control Panel* komputer tersebut.



Gambar 3.33. Microsoft ODBC

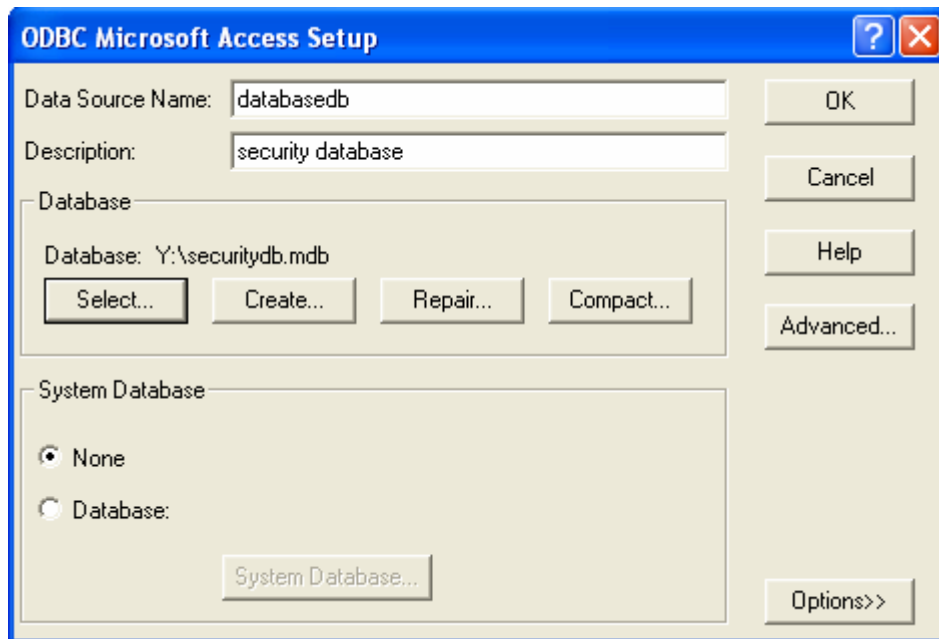
Untuk membuat akses *data sources* baru klik pada “add” kemudian akan keluar tampilan seperti di bawah ini.



Gambar 3.34. ODBC Driver Database

Untuk membuat Data Sources baru, harus dipilih *driver* dari software *database* yang digunakan. *Database* yang dibuat di tugas akhir ini menggunakan Microsoft Access.

Kemudian pada *ODBC Microsoft Access Setup*, diketik nama *database* yang di akses serta *select file data source*-nya yang ada di *server network* seperti yang terlihat pada gambar di bawah ini.



Gambar 3.35. ODBC Setup

Setelah itu database yang terdapat pada server sudah dapat digunakan untuk aplikasi Wonderware InTouch yang terdapat pada *client*.