

## BAB III

### PERANCANGAN DAN IMPLEMENTASI SISTEM

#### 3.1. PERANCANGAN MODEL SISTEM

##### 3.1.1. Penentuan Komponen Dasar

Sistem teleoperasi yang akan dirancang pada penelitian ini berbasis jaringan komputer sehingga untuk komponen dasarnya bisa ditentukan sebagai berikut :

- Pengendali sisi jauh menggunakan *Personal Computer*. Koneksi ke obyek yang dikendalikan dengan memakai perantara rangkaian antarmuka dan rangkaian *buffer*. Komputer yang digunakan sebagai pengendali sisi jauh tidak memerlukan spesifikasi khusus selama sanggup menjalankan *Server Application Program* serta bisa mengidentifikasi rangkaian antarmuka.
- Pengendali lokal menggunakan *Personal Computer*. Komputer yang digunakan sebagai pengendali lokal juga tidak memerlukan spesifikasi khusus selama sanggup menjalankan *Client Application Program*.
- Media transmisi menggunakan jaringan komputer lokal. Pemakaian topologi tidak mempengaruhi sistem selama komputer pengendali sisi jauh sanggup merespon komputer pengendali lokal.

##### 3.1.2. Penentuan Klasifikasi Sistem

Sistem yang akan dirancang hanya bekerja secara satu arah (*open loop*) sehingga operator pengendali lokal belum bisa mengidentifikasi apabila peralatan

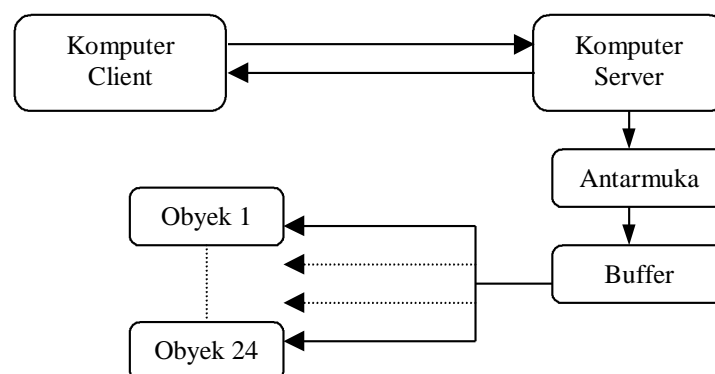
yang dikendalikannya sudah sesuai dengan apa yang diinginkan, Untuk mengatasi hal tersebut serta agar lebih mempermudah komunikasi antar operator, program dilengkapi dengan fasilitas *chatting* sehingga operator pada pengendali lokal bisa berkomunikasi dengan operator pengendali sisi jauh.

### 3.1.3. Penentuan Obyek

Pada perancangan ini, obyek yang akan dikendalikan berupa peralatan elektronik yang tegangan kerjanya 220 V seperti televisi, radio, lampu, kipas angin, *air conditioner* dan peralatan elektronik lainnya dengan batasan masalah menghidupkan-mematikan (*ON-OFF*) peralatan elektronik tersebut. Adapun jumlah obyek yang dapat dikendalikan maksimal 24 obyek

### 3.1.4. Pembuatan Blok Sistem

Setelah ditentukan komponen dasar, cara kerja sistem serta obyek maka dapat dibuat blok sistem yang akan dibuat :



Gambar 3.1. Blok Sistem

## 3.2. PERANCANGAN PERANGKAT KERAS

### 3.2.1. Perancangan Rangkaian Antarmuka

Seperti yang sudah dijelaskan pada bab II, untuk perancangan antarmuka, dipilih antarmuka PPI 8255 karena kemudahan dalam pemrogramannya serta mempunyai 24 bit I/O port sehingga sanggup mengendalikan 24 obyek dengan asumsi 1 obyek memakai 1 bit I/O.

#### 3.2.1.1. Penentuan *mode* / Protokol Komunikasi

Untuk mode/protokol komunikasi, ditentukan dengan memakai *Mode 0* (*Simple protokol*) / *Basic Input-Output* karena sudah mencukupi untuk perancangan ini.

#### 3.2.1.2. Penentuan *Control Word*

Semua *port* akan diinisialisasi sebagai *output*. Dari tabel 2.1, dapat ditentukan *control word* yang akan digunakan yaitu 128 (*Port A, Port B dan Port C sebagai output*).

#### 3.2.1.3. Penentuan *Base Address*

Sedangkan untuk *Base Address*, dari lampiran Peta Alamat I/O PC-XT digunakan alamat 280h – 2EFh. Cakupan alamat ini mengandung 111 alamat yang mungkin. Untuk modul PPI 8255 dibutuhkan 4 buah alamat, jadi bisa digunakan :

*Base Address* = 280h = 10 10000000

*Port A = Base Address* = 280h = 10 10000000

$$\text{Port B} = \text{Base Address} + 1 = 281\text{h} = 10\ 10000001$$

$$\text{Port C} = \text{Base Address} + 2 = 282\text{h} = 10\ 10000010$$

$$\text{Control word} = \text{Base Address} + 3 = 283\text{h} = 10\ 10000011$$

Bobot paling rendah, bit A0 dan A1 dihubungkan langsung dengan masukan alamat pada PPI 8255, sisanya harus di-*decode*. Hal ini dapat dilakukan dengan memakai pembanding (*komparator*) 8 bit. Jadi alamat yang bersangkutan pada *komparator* diperbandingkan dengan alamat seharusnya. Jika sama (P=Q), maka PPI 8255 diaktifkan (CS aktif).

Pengaturan *base address* bisa dilakukan dengan mudah pada *PPI 8255 Card*. Pada modul PPI 8255 yang digunakan pada perancangan ini memakai *dip saklar* dengan tingkat “*high*” atau “*low*” sehingga pemilihan alamat modul PPI 8255 dapat diatur melalui saklar ini.

Tidak digunakannya alamat 300h-31F yang lazim digunakan sebagai alamat *expansion slot modul* semata-mata untuk mengurangi kemungkinan konflik apabila kelak akan ada penambahan *expansion slot modul* lagi pada komputer *server*. Rangkaian lengkap antarmuka terdapat pada lampiran.

### 3.2.2. Perancangan Rangkaian *Buffer*

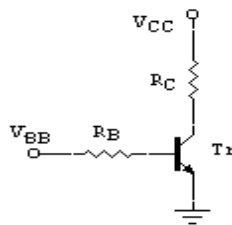
Seperti yang sudah dijelaskan pada BAB II, komputer hanya menggunakan operasi dua keadaan yaitu operasi 1 (*high* atau setara 5 V) dan operasi 0 (*low* atau setara 0 V). Dari sini, untuk perancangan yang akan dibuat, maka bisa dibuat suatu rumusan sebagai berikut :

- Apabila terjadi operasi 1, maka peralatan elektronik akan hidup (*ON*),
- Dan apabila terjadi operasi 0, maka peralatan elektronik akan mati (*OFF*).

Untuk mempermudah penjelasan, rangkaian *buffer* diberi nama *dP-Buffer*

### 3.2.2.1. Penentuan Tipe Bias

Seperti yang sudah dijelaskan pada bab II, salah satu fungsi dari transistor adalah sebagai saklar, maka perancangan rangkaian *dP-buffer* akan menggunakan transistor sebagai saklar digital untuk menghidupkan relay. Sesuai dengan tabel 2-2 tentang tipe-tipe bias, maka tipe yang umum digunakan sebagai saklar adalah bias basis. Maka bisa dirancang rangkaian sebagai berikut :



Gambar 3.2. Bias Basis

### 3.2.2.2. Penentuan Garis Beban dc

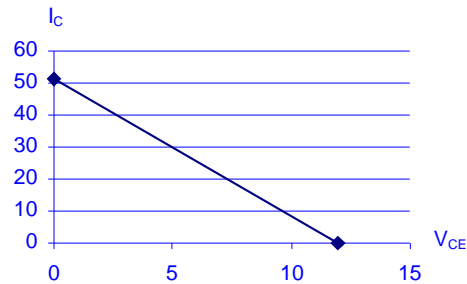
$R_C$  ditentukan sebesar  $33 \Omega$ , kemudian transistor dipilih tipe CS 9014 yang memiliki  $\beta_{dc}$  sebesar 60. Untuk perancangan, harus dihitung nilai  $R_B$  yang titik  $Q$ -nya terletak pada garis beban dc. Dengan memakai persamaan (2-5), dicari  $I_{C(sat)}$  terlebih dulu :

$$I_{C(sat)} \cong \frac{V_{CC}}{R_C} = \frac{12}{33 + 200} = 51,50mA$$

$R_C$  didapat dari Resistor kolektor yang di seri dengan relay yang mempunyai impedansi  $200 \Omega$ . Kemudian dari persamaan (2-4) :

$$V_{CE(cutoff)} = V_{CC} = 12V$$

Dari  $I_{C(sat)}$  dan  $V_{CE(cutoff)}$  bisa digambar garis beban DC :



Gambar 3.3. Garis Beban Transistor Buffer

### 3.2.2.3. Penentuan Daerah Aktif (*Active Region*)

Untuk menentukan titik  $Q$ , harus ditentukan nilai  $R_B$  minimal. Dari persamaan (2-6) :

$$I_B = \frac{V_{BB} - V_{BE}}{R_B} = \frac{4,5 - 0,7}{4.427,20} = 858,33 \mu A$$

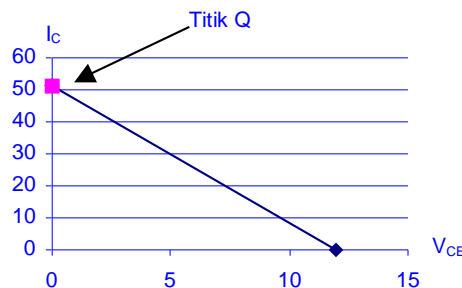
dari persamaan (2-2) :

$$I_C = \beta_{dc} \cdot I_B = 60 \cdot 858,33 = 51,4998 \text{ mA}$$

$$V_{CE} = V_{CC} - I_C R_C = 12 - 51,4998(10^{-3}) \cdot 233 = 0,00005466V$$

Dimana  $V_{BB}$  adalah input yang berasal dari PPI 8255 dan  $\beta_{dc}$  didapat dari tabel persamaan transistor. Dari hasil di atas bisa dilihat letak titik  $Q$  apabila dipasang

$R_B$  minimal :



Gambar 3.4. Garis Beban  $R_B$  minimal

Titik  $Q$  hampir terletak di sumbu  $Y$ .  $R_B$  harus mempunyai nilai yang lebih tinggi dari  $4.427,20 \Omega$ . Misalnya  $R_B$  ditentukan  $10K\Omega$ , maka :

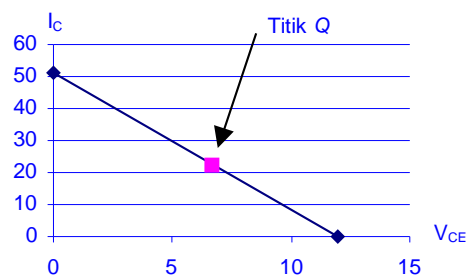
$$I_B = \frac{V_{BB} - V_{BE}}{R_B} = \frac{4,5 - 0,7}{10(10^3)} = 380 \mu A$$

$$I_C = \beta_{dc} \cdot I_B = 60 \cdot 380 \mu A = 22,8 \text{ mA}$$

$$V_{CE} = V_{CC} - I_C R_C = 12 - 22,8(10^{-3}) \cdot 233 = 6,68 \text{ V}$$

$$I_{B(sat)} = \frac{I_{C(sat)}}{\beta_{dc}} = \frac{51,50}{60} = 858 \mu A$$

Sehingga letak titik  $Q$  :



**Gambar 3.5. Garis Beban  $R_B$  10K**

Dari grafik terlihat bahwa  $R_B=10K$ , titik  $Q$  terletak pada garis beban dc. Selanjutnya untuk rangkaian  $dP$ -Buffer digunakan  $R_B=10 K$ .

#### 3.2.2.4. Perancangan *Printed Circuit Board* (PCB)

Terdapat dua buah PCB KIT yang dipergunakan yaitu PCB KIT 8255 dan PCB KIT  $dP$ -Buffer. Untuk PCB KIT 8255, bisa didapatkan dengan mudah di luar sehingga tidak diperlukan perancangan dan pembuatan PCB KIT 8255. Sedangkan untuk PCB KIT  $dP$ -Buffer, karena tidak bisa didapatkan di luar maka harus dilakukan perancangan PCB KIT  $dP$ -Buffer.

*Lay out* untuk PCB KIT *dP-Buffer* dirancang secara manual, dilanjutkan dengan pelarutan PCB yang telah dirancang *layout*-nya, kemudian pengeboran titik-titik komponen dan diakhiri dengan penyolderan komponen dan perakitan *case* untuk PCB tersebut. Rangkaian lengkap *dP-Buffer* terdapat pada lampiran.

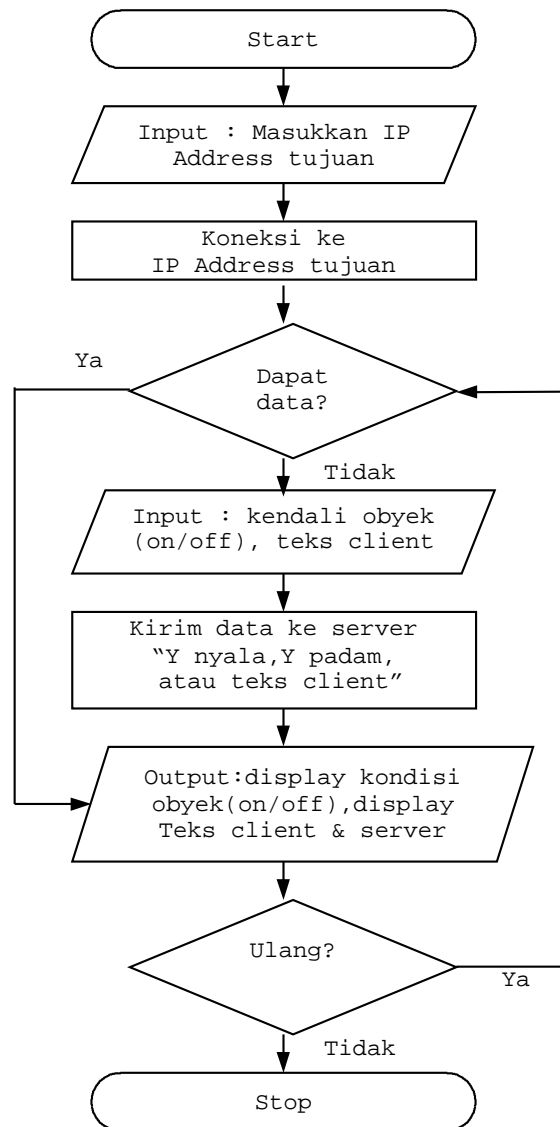
### **3.3. PERANCANGAN PERANGKAT LUNAK**

#### **3.3.1. Perancangan Alur Program.**

Pada perancangan perangkat lunak, maka hal yang paling penting adalah pembuatan alur program untuk komputer *client* dan komputer *server*. Untuk memudahkan, program untuk komputer *client* diberi nama *dP-Client*, sedang program untuk komputer *server* diberi nama *dP-Server*.

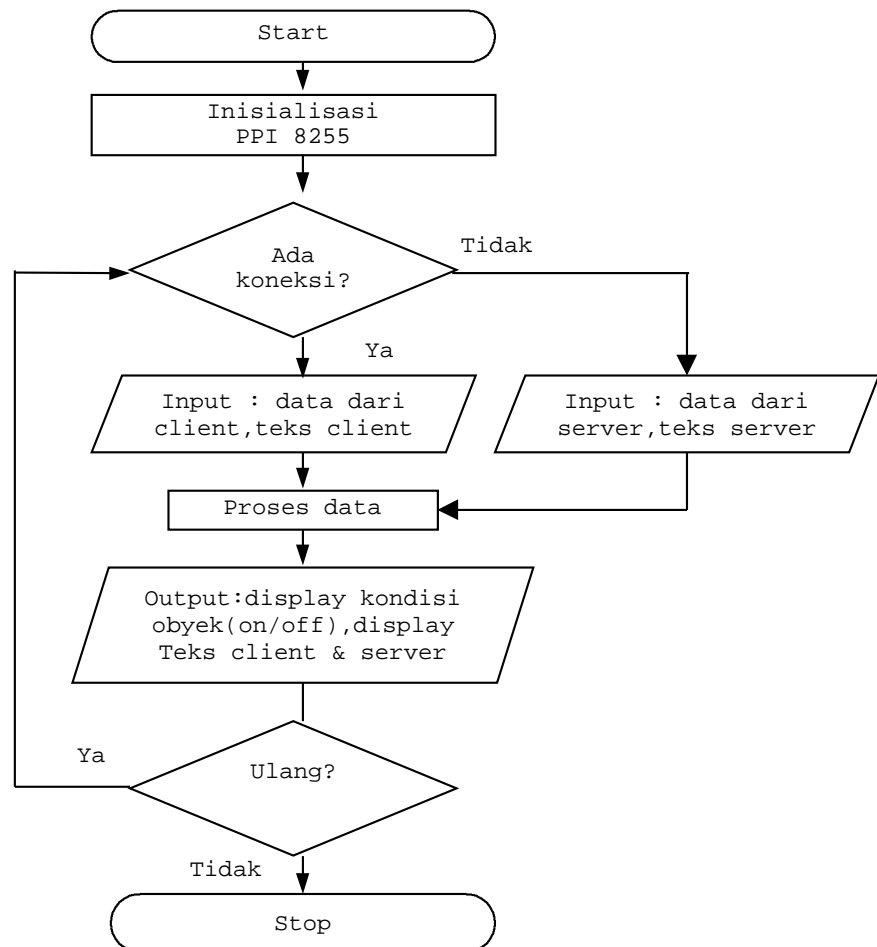
Adapun alur program utama untuk *dP-Client* adalah sebagai berikut, sedang alur program lengkap beserta *sources code dP-Client program* terdapat pada lampiran.





**Gambar 3.6. Alur Program Utama *dP-Client***

Sedang alur program utama untuk *dP-Server* adalah sebagai berikut, sedang alur program lengkap beserta *sources code dP-Server program* terdapat pada lampiran.



**Gambar 3.7. Alur Program Utama dP-Server**

### 3.3.2. Penentuan Topologi, protokol, IP Address, Port Number

Pemakaian topologi apapun (*Linear Bus Topology*, *Ring Topology* atau *Star Topology*) tidak mempengaruhi program dan sistem selama komputer *server* sanggup merespon komputer *client*.

Protokol jaringan memakai *TCP/IP Protocol* karena sudah diterima luas dan praktis menjadi standar de-facto jaringan komputer saat ini.

Pemakaian kelas apapun (kelas A, kelas B dan kelas C) tidak mempengaruhi program dan sistem selama komputer *server* sanggup merespon komputer *client*. Untuk *Port Number* dipilih 17679.

### 3.4. PENGUJIAN SISTEM

Setelah dilakukan perancangan sistem, tibalah saatnya untuk melakukan pengujian terhadap sistem yang telah dibuat. Adapun untuk pengujian awal hanya digunakan komputer tunggal / *stand alone computer* sehingga program *dP-Client* hanya di *loopback* ke program *dP-server*. Pada pengujian tahap ini, sistem berfungsi sebagaimana mestinya. Komputer yang dijadikan pengujian juga tidak menunjukkan kinerja yang abnormal. Spesifikasi komputer pada pengujian tahap awal ini menggunakan :

- Intel Pentium 233 MHz
- Memory 32 M

Untuk *video card*, *harddisk*, *CD-ROM*, *motherboard* memang tidak dicantumkan karena tidak berpengaruh terhadap sistem ini. Rangkaian *dP-Relay* dihubung ke modul PPI 8255 dengan memakai kabel *data 25 bus*, sedangkan modul PPI 8255 dipasang pada slot *ISA*. Pemasangan diatur sedemikian rupa hingga tidak mengganggu *card-card* yang lain.

Dari pengujian awal, dilanjutkan ke pengujian dengan memakai jaringan komputer. Digunakan komputer *client* dengan spesifikasi sebagai berikut :

- AMD T-Bird 900 MHz
- Memory 128 M

- Lan Card Surecom

Sedangkan komputer *server* dengan spesifikasi sebagai berikut :

- Intel Pentium 233 MHz
- Memory 32 M
- Lancard Surecom

Untuk *video card, harddisk, CD-ROM, motherboard* memang tidak dicantumkan karena tidak berpengaruh terhadap sistem ini. Sedangkan untuk jaringan :

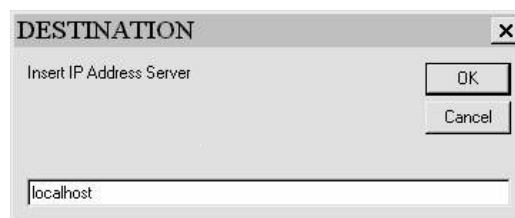
- *concentrator* berupa *hub 8 port 10 M*
- Topologi jaringan memakai *Star Topology*
- Protokol komunikasi memakai *TCP/IP*
- *IP Address* digunakan kelas A dari 10.0.0.1 sampai 10.0.0.6 karena jumlah *host* yang terhubung ke jaringan hanya 6 komputer.

Dari pengujian ini, ketika program *dP-server* belum dijalankan pada komputer *server*, program *dP-Client* yang dijalankan pada komputer *client* menjalankan prosedur *error*. Ketika *dP-Server* sudah dijalankan, operator *client* tanpa harus login dan operator *server* bisa saling berkomunikasi dengan jalan *chatting*. Operator *client* harus memasukkan LOGIN-ID dan PASSWORD-ID apabila ingin mengendalikan obyek. Obyek yang aktif ditandai dengan *frame* hijau, sedang obyek yang non aktif ditandai dengan *frame* merah.

Saat komunikasi sedang berlangsung, apabila ada *client* lain yang ingin mengakses *server*, harus menunggu sampai *client* sebelumnya memutuskan hubungannya dengan *server* karena sistem dirancang untuk *single user*.

Program *dP-Server* harus selalu *stand-by*, dengan kata lain komputer *server* harus selalu *on*. Dan komputer-komputer *server* memang dirancang untuk selalu *stand-by*.

Ketika program *dP-Client* dijalankan, program akan meminta nomor *IP Address* tujuan. Dalam hal ini adalah nomor *IP Address* komputer *server*, komputer di mana rangkaian antarmuka dan *dP-buffer* dipasang. Apabila Operator *Client* tidak mengisi maka dianggap memakai *IP Loopback* (127.0.0.1).



**Gambar 3.8. Tampilan Destination**

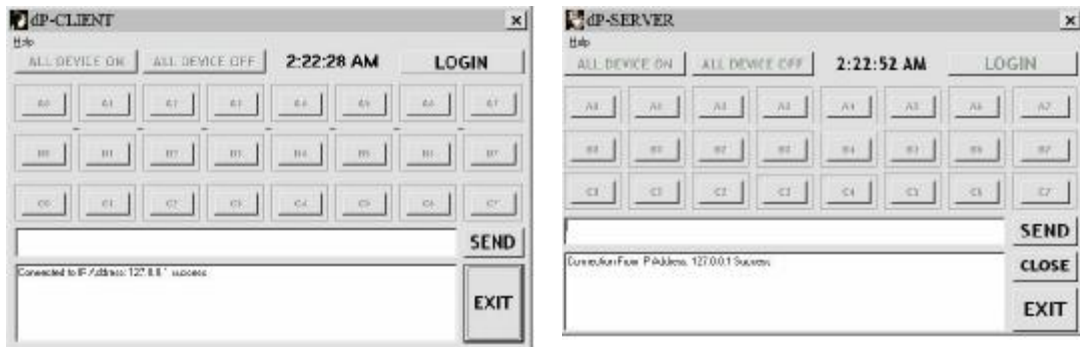
Setelah Operator *Client* mengisi *IP Address*, maka program akan mencoba koneksi ke alamat tujuan dengan memakai nomor *port* yang sudah ditentukan yaitu 17679. *Server* juga memakai nomor *port* ini sehingga komunikasi antara *client* dan *server* bisa terwujud. Apabila koneksi tidak berhasil sebab komputer *server* belum siap atau salah memasukkan *IP Address* komputer *server*, maka prosedur *connection error* akan dijalankan.



**Gambar 3.9. Tampilan Connection Failure**

Apabila koneksi berhasil maka Operator *Client* masuk ke tampilan program *dP-Client*. *Text display* menunjukkan status bahwa program telah terkoneksi ke

*Server* (dalam contoh di atas ke alamat *loopback* 127.0.0.1) dan begitu juga pada tampilan pada *dP-Server*.



**Gambar 3.10. Tampilan *dP-Client* dan *dP-server* saat koneksi berhasil**

Pertama kali masuk ke tampilan program *Client*, semua tombol non-aktif kecuali LOGIN, SEND dan EXIT. Artinya Operator *Client* apabila ingin menjalankan program harus memasukkan *Login-ID* beserta *Password-ID*. Tombol *send* aktif, memungkinkan operator pada sisi *client* bisa berkomunikasi dengan operator sisi *server* tanpa harus *Login*.

Tampilan *Login-ID* dan *Password-ID* sebagai berikut :



**Gambar 3.11. Tampilan *dP-Client Login* dan *dP-Client Password***

Apabila *Login-ID* dan *Password-ID* salah, maka muncul tampilan *wrong-ID* :



**Gambar 3.12. Tampilan *Wrong-ID***

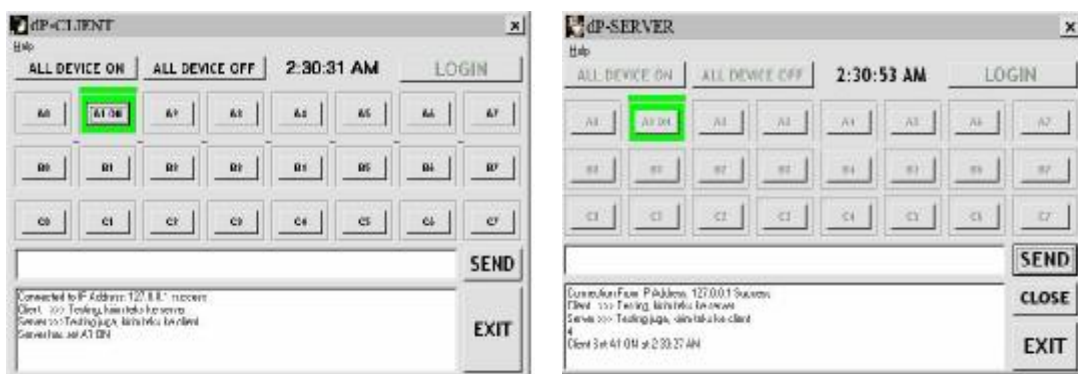
Apabila *Login-ID* benar dan *Password-ID* salah, maka muncul tampilan *Password-ID invalid* :



Gambar 3.13. Tampilan *Password-ID* invalid

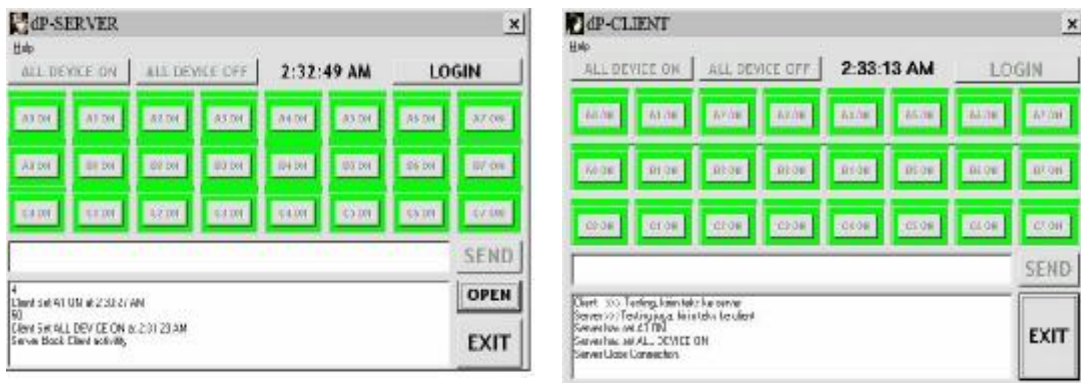
Prosedur *wrong-ID* serta prosedur *Password-ID* *invalid* merupakan prosedur *looping* tanpa batas. Artinya ketika operator *client* belum memasukkan *Login-ID* dan *Password-ID* yang benar, operator *client* hanya dihadapkan pada tampilan *dP-Client* yang non-aktif.

Ketika *Login-ID* dan *Password-ID* benar, operator *client* bisa mengakses obyek yang terpasang pada sistem. Pada perancangan ini, operator *client* bisa mengendalikan sampai 24 obyek. Saat suatu obyek diakses, dilengkapi dengan jam pengaksesan sehingga operator pada kedua sisi mengetahui kapan obyek tersebut terakhir kali diakses. Juga dilengkapi dengan *chatting mode*, sehingga Operator *client* juga bisa berkomunikasi dengan operator *server* dengan menulis *text* pada *textbox*.



Gambar 3.14. Tampilan saat *client* mengakses obyek

*Server* mempunyai hak untuk menutup koneksi dari *client*. Sehingga ketika koneksi ditutup oleh *server*, *client* harus keluar dari program serta *Login* ulang untuk masuk ke sistem.



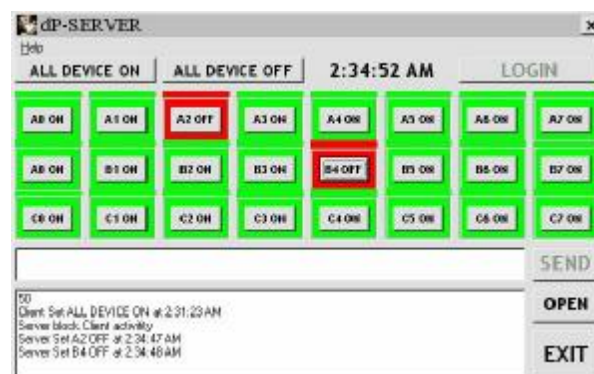
**Gambar 3.15. Tampilan saat Server menutup koneksi**

Apabila *server* ingin mengakses obyek, harus memberikan *Password-ID* :



**Gambar 3.16. Tampilan dP-Server Password**

Setelah *password* sukses baru *server* bisa mengakses obyek.



**Gambar 3.17. Tampilan dP-Server setelah dP-Server Password sukses**



Apabila *server* membuka koneksi, maka *client* bisa untuk mengakses kembali obyek. Saat *Server* membuka koneksi, maka semua tombol non-aktif. Hal ini dimaksudkan agar tidak terjadi bentrok pengaksesan obyek antara *Server* dan *Client*.



**Gambar 3.18. Tampilan saat *Server* membuka koneksi kembali**

Program *dP-Server* dirancang untuk bekerja secara *single-user* sehingga apabila *server* sedang koneksi dengan *client A*, maka *client* lainnya tidak bisa koneksi ke *server* dan prosedur *connection error* tampil. Dengan demikian tidak mungkin terjadi bentrokan akses antara *client* yang satu dengan *client* lain atau bentrokan akses antara *client* dengan *server*.