

# PENGEMBANGAN APLIKASI REMOTE SPESIFIKASI DESKTOP BERBASIS CLIENT SERVER

<sup>(a)</sup>Herlino Nanang, <sup>(b)</sup>Andrew Fuadi, <sup>(c)</sup>Nurul Farhanah,

<sup>(a)</sup>Staf Pengajar Sains dan Teknologi  
Universitas Islam Negeri Syarif Hidayatullah Jakarta

<sup>(b)</sup>Staf Pengajar Sains dan Teknologi  
Universitas Islam Negeri Syarif Hidayatullah Jakarta

<sup>(c)</sup>Mahasiswa Fakultas Sains dan Teknologi  
Universitas Islam Negeri Syarif Hidayatullah Jakarta

## ABSTRAK

Dalam sebuah jaringan komputer terdapat keinginan untuk mengorganisir jaringan secara terpusat berbasis client server, mengetahui spesifikasi komputer, penggunaan cpu, alokasi memori yang digunakan dan hardisk yang tersedia pada tiap komputer, dan mengendalikan client oleh servernya. Sehingga timbul masalah bagaimana cara spek komputer client dapat dikirim ke server, bagaimana mengendalikan komputer client dan memonitoring komputer client dan diharapkan menghasilkan aplikasi yang membantu dalam penghematan waktu dan tenaga dalam mengontrol aktivitas user pada jaringan. Maka dibuat aplikasi sederhana untuk melakukan pengecekan terhadap spesifikasi komputer client, pengendalian, dan pengawasan terhadap komputer yang berada pada jaringannya menggunakan bahasa pemrograman Delphi 7.0. Metodologi penelitian yang digunakan dalam pembuatan aplikasi ini adalah Rapid Application Development. Aplikasi ini dapat berjalan pada system operasi windows.

Kata Kunci : Client Server, Speksifikasi komputer, remote.

## 1. PENDAHULUAN

### 1.1 Latar belakang masalah

Akses terhadap komputer yang berada dalam satu jaringan dapat berupa pengaksesan terhadap file ataupun aplikasi. Termasuk pula pengaksesan dalam spesifikasi komputer yang berada pada jaringannya. Akses terhadap spesifikasi komputer dalam satu jaringan bisa melalui pengontrolan atau pengendalian host-host dengan menggunakan aplikasi *remote monitoring* atau *remote control* komputer. Namun, *remote monitoring* pun memiliki keterbatasan dalam mengetahui spesifikasi komputer secara langsung.

Keinginan server untuk dapat mengorganisir jaringan secara terpusat, mengetahui spesifikasi komputer, penggunaan cpu, alokasi memori yang digunakan dan *hardisk* yang tersedia pada tiap komputer, keinginan untuk dapat berkomunikasi, serta keinginan untuk dapat mengendalikan serta mengontrol aktifitas *client* merupakan alasan dibuatnya aplikasi yang dapat memantau sekaligus mengetahui spesifikasi komputer pada *client*.

Aplikasi ini merupakan penggabungan dari konsep aplikasi *remote monitoring* dan aplikasi pendeteksi spesifikasi komputer yang telah ada sebelumnya. Dengan adanya aplikasi ini, diharapkan

dapat mengefisiensikan waktu dalam melakukan pengecekan spesifikasi komputer *client*, sekaligus dapat memantau serta mengendalikan aktivitas yang sedang dilakukan oleh *client*.

### 1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan tersebut, dapat dirumuskan masalah yang harus dipecahkan yaitu :

1. Bagaimana menampilkan keterangan spesifikasi komputer *client* yang di kirim ke komputer *server*?
2. Bagaimana *server* dapat me-remote komputer *client*?
3. Aktivitas *remote* apa saja yang dapat dilakukan oleh *server*?

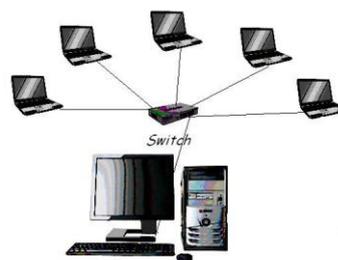
### 1.3 Batasan Masalah

Untuk menghindari meluasnya ruang lingkup yang dibahas pada penelitian ini, maka diberikan batasan masalah sebagai berikut:

1. Percobaan terhadap aplikasi ini hanya akan dilakukan dalam sebuah jaringan LAN.

2. Aplikasi ini terbagi 2, yaitu aplikasi *client* dan aplikasi *server*.
3. Aplikasi dijalankan dalam sistem operasi *Windows*.
4. Perangkat lunak yang akan digunakan untuk membangun aplikasi ini adalah Borland Delphi 7.
5. Antarmuka perangkat lunak akan dikembangkan dalam bahasa Inggris karena alasan banyaknya istilah asing yang bila dialih-bahasakan ke bahasa Indonesia akan menimbulkan kerancuan.
6. Metodologi yang digunakan pada penelitian ini adalah *Rapid Application Development (RAD)*.

- a. Aktif
- b. Mengirim *request*
- c. Menunggu dan menerima balasan dari *server*



Contoh *client server* dengan 5 *client* 1 *server*

#### 1.4 Manfaat

Diharapkan aplikasi ini dapat berguna bagi masyarakat luas.

#### 1.5 Tujuan

Tujuan yang ingin dicapai oleh penulis dari penelitian ini adalah :

1. Menghasilkan sebuah aplikasi *remote LAN* yang dapat mengirim data spesifikasi komputer *client* yang akan di tampilkan di aplikasi *server*.
2. Dengan aplikasi ini diharapkan *server* dapat memantau aktivitas *user* melalui *remote monitoring client* yang merupakan dasar dari pengembangan aplikasi *remote* ini.

## 2. LANDASAN TEORI

### 2.1 Client Sever

*Client-server* yaitu jaringan komputer dengan komputer yang didedikasikan khusus sebagai *server*. Sebuah *service*/layanan bisa diberikan oleh sebuah komputer atau lebih.

*Server* adalah komputer yang dapat memberikan *service* ke *client*, sedangkan *client* adalah komputer yang mengakses beberapa *service* yang ada di *server*. Ketika *client* membutuhkan suatu *service* yang ada di *server*, dia akan mengirim *request* kepada *server* lewat jaringan. Jika *request* tersebut dapat dilaksanakan, maka *server* akan mengirim balasan berupa *service* yang dibutuhkan untuk saling berhubungan menggunakan *socket*.

1. Karakteristik *Server*
  - a. Pasif
  - b. Menunggu *request*
  - c. Menerima *request*, memproses mereka dan mengirimkan balasan berupa *service*
2. Karakteristik *Client*

*Client/Server* merupakan teknologi pendistribusian kerja aplikasi antara dua komputer atau lebih, yang dihubungkan oleh jaringan komunikasi, dimana yang satu akan bertindak sebagai *client* atau peminta layanan, dan yang lainnya sebagai *server*, atau pemberi layanan. Baik *client* ataupun *server* memiliki pemroses atau CPU sendiri, sedangkan jaringan yang digunakan bisa berupa jaringan lokal (*LAN*) ataupun jaringan yang lebih luas lagi (*WAN*). Dalam model *Client/Server*, sebuah aplikasi dibagi menjadi dua bagian yang terpisah, tapi masih merupakan sebuah kesatuan yakni komponen *client* dan komponen *server*. (<http://mokta-edynurrachyanto.blogspot.com/2010/03/client-server-pada-jaringan.html>)

### 2.2 Internet Direct (Indy)

*Indy* digunakan untuk membuat aplikasi yang menggunakan protocol jaringan. *Indy* adalah komponen *open source* yang terdiri dari protocol internet yang sudah begitu populer, seperti *HTTP*, *FTP*, *SMTP*, *TCP*, *UDP*, *Gopher*, *Whois* dan lain-lain. Karena *open source* maka semua kode program yang ada dapat dikembangkan kembali. Kemudahan *indy* yang lain yaitu mendukung protocol sampai 120 protokol. (pujon online- [www.pherie.my-php.net](http://www.pherie.my-php.net)).

Protokol jaringan adalah *software* yang digunakan untuk menghubungkan komputer dalam jaringan supaya dapat saling berkomunikasi. Protokol terdiri dari aturan-aturan yang dipakai dalam berkomunikasi. Protokol yang banyak digunakan untuk menghubungkan komputer dalam jaringan, antara lain *ATM*, *Apple Talk*, *DCL*, *Net BEUI*, *NWLink* *IPX/SPX*, *TCP/IP*, dan lain-lain.

Komponen internet *direct* (*indy*) digunakan untuk aplikasi jaringan berbasis *Client Server*. Aplikasi jaringan yang dibuat dengan komponen *indy* terhubung dalam sebuah jaringan *Local Area Network (LAN)*, komponen *indy* terbagi menjadi dua group utama yaitu *indy server* dan *indy client* dimana *indy server* ditempatkan pada aplikasi *server* sedangkan *indy client* ditempatkan pada aplikasi *client*.

Internet langsung (*Indy*) adalah seperangkat komponen *Internet open source*, meliputi hampir semua protokol *Internet* yang populer. Ditulis dengan

Delphi Indy, yang termasuk dalam Delphi 6, Kylix 1 dan C++ Builder 6 dan di atas dalam semua versi lingkungan pengembangan Borland.

Komponen dalam internet *direct* atau *indy* antara lain :

1. **IdICMPCClient**  
*Internet Control Message Protocol (ICMP)* adalah protokol yang bertugas mengirimkan pesan-pesan kesalahan dan kondisi lain yang memerlukan perhatian khusus. Pesan atau paket ICMP dikirim jika terjadi masalah pada layer IP dan layer atasnya (TCP/UDP).
2. **TIdIPWatch**  
Komponen ini digunakan untuk mendapatkan informasi *IP Address* yang terdapat dalam komputer yang sedang aktif.
3. **Proxy HTTP**  
*Hypertext Transfer Protocol (HTTP)* merupakan protokol komunikasi yang digunakan untuk men-transfer informasi pada *World Wide Web*. Fungsi originalnya adalah menyediakan cara untuk mem-publish dan menerima halaman *HTML hypertext*. HTTP merupakan protokol *request/response* antara *client* dan *server*. *Client* membuat sebuah *HTTP request*, seperti *web browser*. *Server* – merupakan komputer yang menyimpan atau membuat *file HTML*. Sebuah *client HTTP* akan menginisialisasi sebuah *request* dengan membangun sebuah koneksi TCP ke *port 80* (default) , Sebuah *Server HTTP* akan ‘mendengarkan’ *port*. Saat menerima *request*, *server* akan mengirimkan kembali status seperti : “(HTTP/1.1 200) OK”, dan sebuah *message* untuk dirinya sendiri.
4. **Whois**  
Merupakan sebuah *protocol query* yang berbasis TCP yang digunakan untuk melakukan *query* ke *database* untuk menentukan pemilik dari sebuah domain *name*, alamat IP. Kegunaan dari data yang diberikan oleh WHOIS (hasil *query*), dapat digunakan untuk melakukan validasi untuk *Certificate Authority* misalnya untuk *registrasi ecommerce* dan sebagainya.
5. **TIdUdpServer dan TIdUdpClient**  
*UDP (User Datagram Protocol)* merupakan salah satu protokol lapisan *transport TCP/IP* yang mendukung komunikasi yang tidak andal (*unreliable*), tanpa koneksi (*connectionless*) antara *host-host* dalam jaringan yang menggunakan TCP/IP.

### 3. METODOLOGI PENELITIAN

#### 3.1. Metode Pengumpulan Data

##### 3.1.1 Studi Pustaka

Dalam melakukan pengumpulan data, penulis menggunakan teknik studi pustaka. Pada tahapan pengumpulan data dengan cara studi pustaka, penulis mencari referensi-referensi yang relevan dengan objek yang akan diteliti. Pencarian referensi dilakukan di perpustakaan, maupun secara *online* melalui internet.

##### 3.1.2 Studi Literature

Berdasarkan pengamatan penulis, pada penelitian sebelumnya terdapat beberapa penelitian yang hampir serupa dengan judul Pengembangan Aplikasi *Remote* Spesifikasi Komputer Berbasis *Client Server* antara lain :

1. Rancang Bangun Aplikasi *Remote* untuk Administrasi LAN yang ditulis oleh Ryan Aniestyo Hermoko, Fathul Wahid, M. Andri Setiawan dari Universitas Islam Indonesia. Mereka menyempurnakan aplikasi *Pengendali PC Jarak Jauh pada LAN*. Pada penelitian ini meremote semua aktivitas yang dilakukan oleh *client*.
2. *Literature* aplikasi yang digunakan sebagai acuan adalah aplikasi *remote monitoring* yang digunakan untuk mengendalikan dan memantau aktivitas komputer *client* sesuai nomer IP-nya. Selain itu *literature* aplikasi yang digunakan adalah aplikasi pendeteksi spesifikasi komputer antara lain PC-Wizard, CPU-Z, Everest, Speecy, dan juga DX-Diag. Aplikasi-aplikasi tersebut dapat menampilkan spesifikasi komputer. Kedua *literature* aplikasi ini yang menjadi dasar peneliti mengembangkan aplikasi *remote LAN* yang dapat mengetahui spesifikasi komputer *client* yang terhubung pada *server*.
3. *Net Support Manager* merupakan aplikasi yang dapat mengetahui spesifikasi komputer berbasis *client server*. Aplikasi ini menampilkan beberapa spesifikasi komputer terhadap satu komputer *client* saja. Peneliti mengembangkan aplikasi sejenis yang dapat mengetahui spesifikasi semua komputer *client* yang terhubung sekaligus dapat memantau dan mengendalikan *client*.

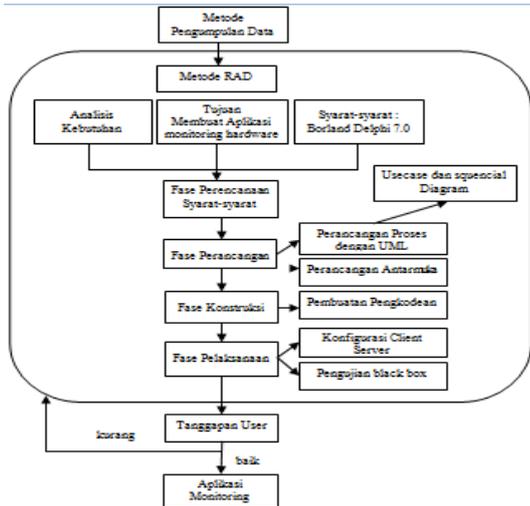
##### 3.2 Metodologi penelitian

Metode pengembangan sistem yang penulis gunakan dalam penelitian ini adalah metode *Rapid Application Development (RAD)*. Pada metode ini terdiri dari 4 tahapan yaitu :

1. Fase Perencanaan Syarat-Syarat : Fase dimana menentukan tujuan dan syarat-syarat yang dibutuhkan untuk memenuhi kebutuhan dalam pengembangan system.
2. Fase Perancangan : Fase dimana perancangan proses-proses yang akan terjadi dalam aplikasi,

perancangan *use case* dan perancangan antarmuka.

- 3 Fase Konstruksi : Pada tahapan ini dilakukan tahap pengkodean terhadap rancangan-rancangan yang telah didefinisikan.
- 4 Fase Pelaksanaan : Pada tahap ini dilakukan pengujian terhadap sistem dan pengenalan terhadap aplikasi.



#### 4. Rancangan sistem

##### 4.1 Fase Perencanaan syarat-syarat

Dalam fase perencanaan ini ditentukan permasalahan yaitu keinginan untuk mengorganisir jaringan secara terpusat berbasis *client server*, mengetahui spesifikasi komputer, penggunaan cpu, alokasi memori yang digunakan dan *hardisk* yang tersedia pada tiap komputer, dan mengendalikan dan *monitoring client* oleh *servernya*.

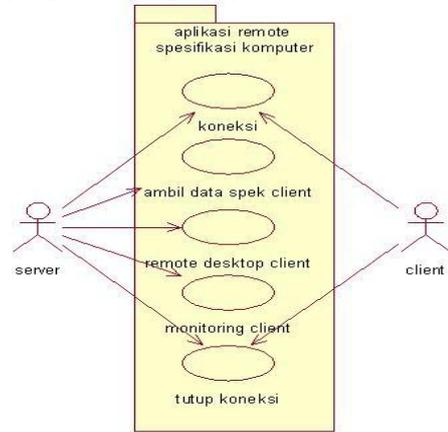
Syarat-syarat yang dibutuhkan untuk dapat membuat aplikasi sesuai yang diinginkan diatas, maka yang dibutuhkan adalah pemilihan bahasa pemrograman untuk pembuatan aplikasi yaitu Delphi 7.0, tambahan library NMM untuk *remote control* dan *alite skin* untuk tampilan.

Selain itu, kita menentukan *port* yang digunakan pada aplikasi ini. Dalam pembuatan aplikasi ini *port* yang akan digunakan adalah *port "dynamic"* atau "*unprivileged*" *port* yaitu *Port* dengan *range* 1024 s.d 65535. Pemilihan *port* ini karena *port* ini tidak banyak digunakan secara umum pada jaringan contohnya berbagai aplikasi yang khas, seperti telnet, mail, web, ftp, dan sebagainya.

##### 4.2 Fase Perancangan

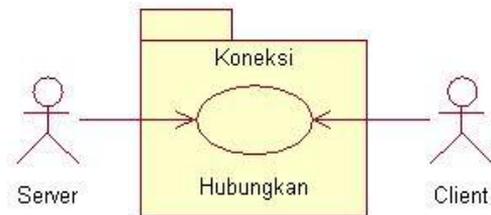
###### 4.2.1 Use case Diagram

*Use Case Diagram* digunakan untuk menjelaskan apa yang akan dilakukan oleh sistem serta aktor-aktor yang akan berhubungan dengan proses-proses yang ada pada aplikasi. *Use case* ini diperlukan untuk kebutuhan dokumentasi dan arah pengembangan selanjutnya. Dibawah ini adalah *use case* untuk beberapa aplikasi yang diinginkan.

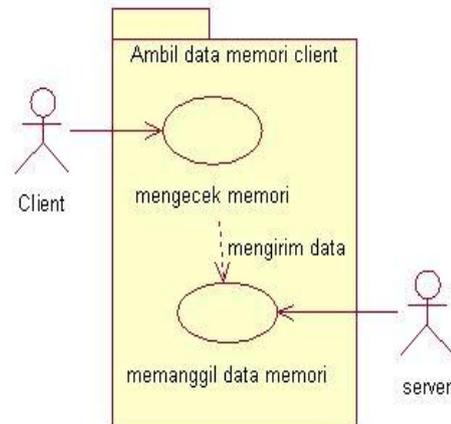


Use case Diagram untuk Aplikasi remote spesifikasi komputer

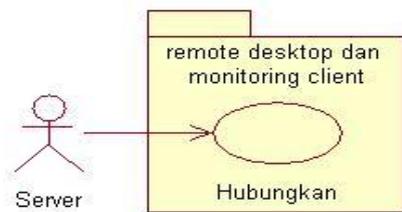
1. Use case Diagram untuk koneksi



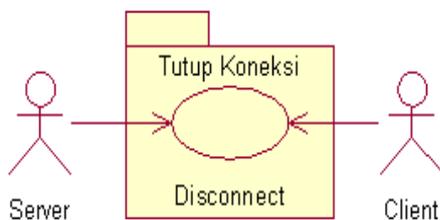
2. Use case Diagram untuk ambil data memori client



3. Use case Diagram untuk remote desktop dan monitoring client



4. Use case Diagram untuk tutup koneksi



#### 4.2.2 Use case Scenario

Use case scenario merupakan penjelasan yang lebih terperinci mengenai masing-masing use case yang terjadi di dalam sistem. Use case scenario ini terdiri dari:

1. Nama use case adalah nama use case yang akan dideskripsikan.
2. Aktor yang terlibat.
3. Trigger.
4. Precondition yang penting bagi use case untuk memulai.
5. Action.
6. Postcondition yang menjelaskan state dari sistem setelah use case berakhir.

Setelah menjelaskan use case pada bahasan sebelumnya, maka berikut ini akan dijelaskan spesifikasi use case yang telah ditentukan.

##### a. Use case Koneksi

|                            |                                      |
|----------------------------|--------------------------------------|
| <b>Nama Usecase</b>        | Koneksi                              |
| <b>Aktor yang terlibat</b> | Server, Client                       |
| <b>Trigger</b>             | Aktor ingin saling terhubung         |
| <b>Pre condition</b>       | IP address dan port telah terdeteksi |
| <b>Action</b>              | Hubungkan                            |
| <b>Post condition</b>      | Terhubung, IP aktif                  |

##### b. Use case Tutup Koneksi

|                            |  |
|----------------------------|--|
| <b>Nama Use case</b>       | Tutup koneksi  |
| <b>Aktor yang terlibat</b> | Server, Client                                       |
| <b>Trigger</b>             | Salah satu atau kedua aktor ingin memutuskan koneksi |
| <b>Pre condition</b>       | Kedua aktor telah terhubung                          |
| <b>Action</b>              | Tutup koneksi  |
| <b>Post condition</b>      | Disconnect, IP tidak aktif                           |

##### c. Ambil data memori client

|                            |  |
|----------------------------|--|
| <b>Nama Use case</b>       | ambil data memori client   |
| <b>Aktor yang terlibat</b> | Server, Client   |
| <b>Trigger</b>             | Server hendak mengetahui informasi tentang Client  |
| <b>Pre condition</b>       | Aktor telah saling terkoneksi  |
| <b>Action</b>              | Client mengecek spesifikasi dirinya, Server dapat langsung memanggil informasi yang telah di tulis oleh Client |
| <b>Post condition</b>      | Data dari Client dapat ditampilkan   |

##### d. Remote desktop dan monitoring client

|                            |   |
|----------------------------|---|
| <b>Nama Usecase</b>        | Remote desktop dan monitoring client                |
| <b>Aktor yang terlibat</b> | Client, Server                                      |
| <b>Trigger</b>             | Server akan me-remote desktop dan monitoring Client |
| <b>Pre condition</b>       | Aktor telah saling terkoneksi                       |
| <b>Action</b>              | View full Client                                    |
| <b>Post condition</b>      | Menampilkan desktop dari Client                     |

#### 4.2.3 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar aplikasi. Sequence diagram untuk aplikasi remote spesifikasi komputer berbasis client server sebagai berikut :

1. Koneksi

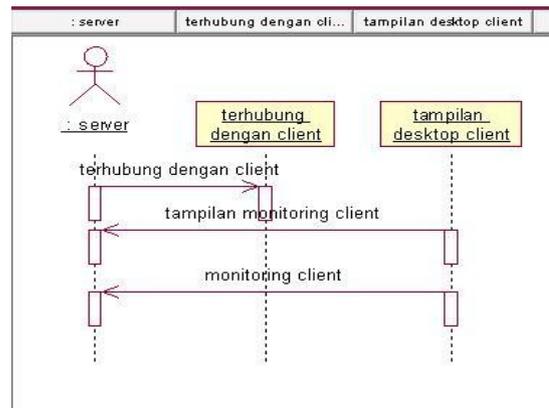
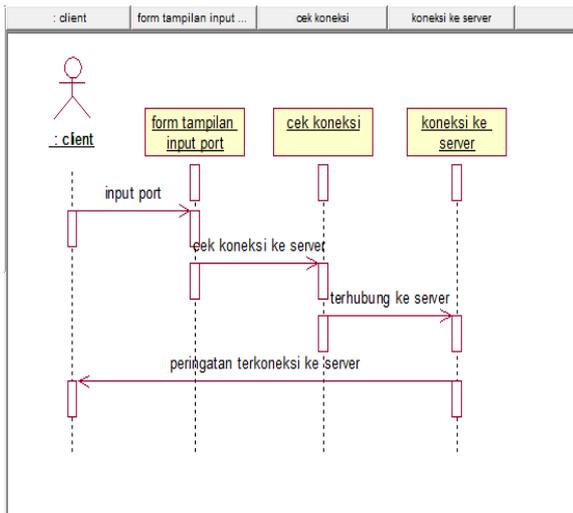
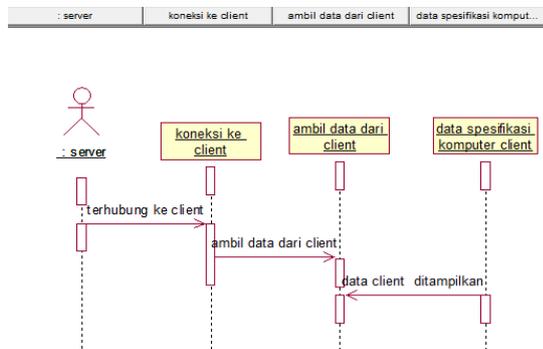


Diagram *sequence* untuk *remote monitoring client* diatas menggambarkan bahwa antara *client* dan *server* saling terkoneksi, maka *server* dapat melakukan *remote monitoring* terhadap *client*. Remote desktop dapat mengendalikan client secara penuh dengan menggunakan *mouse* dan *keyboard server*. Setelah *client* dan *server* saling terkoneksi maka *server* memilih untuk *me-monitoring client* sesuai dengan nomer IP yang akan dipantau, *monitoring* ini untuk mengawasi aktivitas *client*.

Pada diagram *sequence* koneksi, telah terdeteksi IP Address dan *port* ketika aplikasi berjalan. Kemudian *client* meminta supaya dapat terhubung dengan *server*, maka terjadi proses cek koneksi ke *server*. Pengecekan koneksi ini dengan memeriksa *port* dan IP yang sama. Analogi koneksi antara *client* dan *server* ini seperti pendialan nomor telepon yang dituju, apabila telah diijinkan untuk melakukan komunikasi maka terbentuk suatu hubungan.

#### 4. Tutup Koneksi

#### 2. Ambil data spek client



Pada diagram *sequence* ambil data spek *client*, dimana antara *client* dan *server* telah saling terhubung. *Client* mengecek spesifikasi dirinya kemudian *server* dapat memanggil data-data tersebut dari *client* dan hasilnya akan ditampilkan antara lain sistem operasi, resolusi layar, jenis komputer, nama *user*, kapasitas memori, *directory windows* dan sistem32, kecepatan CPU, *BrandID CPU*, serta *vendor CPU*.

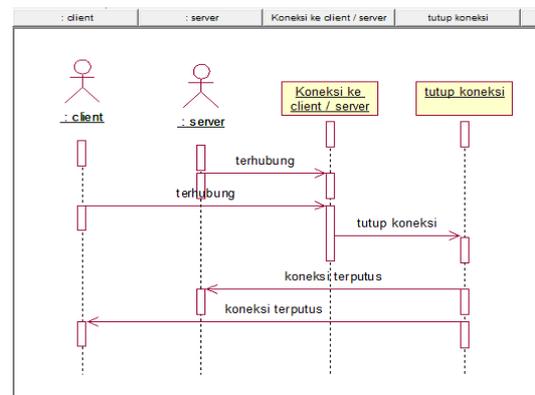
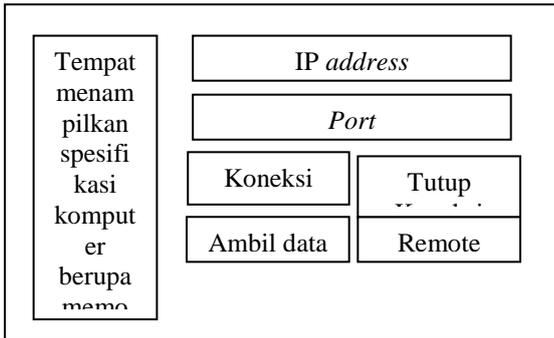


Diagram *sequence* diatas menggambarkan urutan tutup koneksi antara *client* dan *server*. Keduanya dapat melakukan pemutusan koneksi.

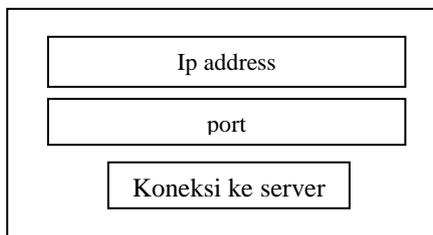
#### 3. Remote monitoring client

#### 4.2.4 Perancangan User interface

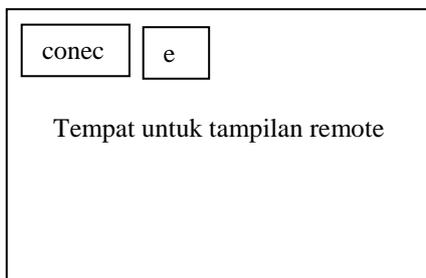
1. Tampilan aplikasi *server*



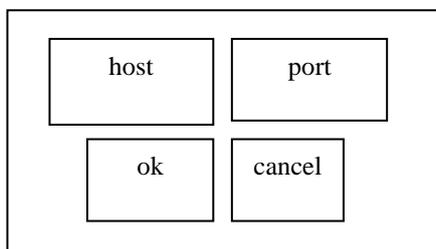
2. Tampilan aplikasi *client*



3. Tampilan *Remote control dan Monitoring*



4. Tampilan koneksi *remote control client*



4.3 Fase Konstruksi

Berdasarkan syarat-syarat kebutuhan dan perancangan untuk pembuatan aplikasi *remote* spek basis *client server* ini, bahasa yang digunakan adalah Delphi 7.0 dengan tambahan komponen NMM dan Alite untuk skin atau tampilan. Instalasi komponen dapat dilihat pada lampiran.

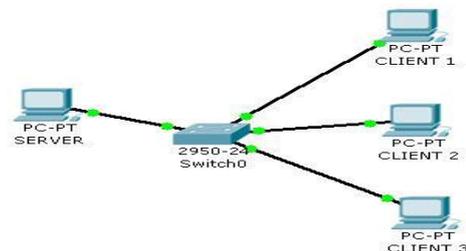
Pembuatan aplikasi ini dibagi menjadi 2 bagian yaitu konstruksi untuk aplikasi *server* dan untuk aplikasi *client*. Konsep yang digunakan untuk menampilkan spek *client* melalui remote pada aplikasi ini konsep *server* menjadi *client* dan sebaliknya *client* menjadi *server*. pada aplikasi ini *server* yang lebih aktif, *client* menunggu *request* dan *server* menerima layanan.

4.4 Fase Pengujian

Pada tahapan pengujian aplikasi *remote* spesifikasi komputer ini, dilakukan dengan cara pengujian mandiri dan *blackbox testing*.

Dalam melakukan pemrograman, penulis melakukan pengujian mandiri yaitu pengujian terhadap kode-kode program untuk memastikan kebenaran program tersebut. Pengujian ini dilakukan untuk mencari kesalahan yang ditimbulkan karena salah tulis atau kesalahan pemrograman. Kegiatan unit *testing* ini disebut juga *debugging*. Kegiatan ini digunakan untuk mencari posisi peringatan (*warning*), kesalahan (*error*) dari kode-kode program dan mengetahui *debugging* yang sukses.

4.4.1 Tipe jaringan yang digunakan



Tipe jaringan ini hanya digambarkan beberapa komputer *client* yang terhubung ke komputer server melalui *switch* atau hub, atau pengujian aplikasi ini dapat melalui jaringan *wireless*.

4.4.2 Spesifikasi computer

Adapun spesifikasi komputer yang digunakan dalam pengujian aplikasi ini adalah :

| N o. | Hard ware      | Server  | Client  |
|------|----------------|---|---|
| 1.   | Proce ssor     | Dual-Core 2.20 GHz  | Pentium 4   |
| 2.   | Mem ory        | 1 GB  | 1 GB  |
| 3.   | Monit or       | resolusi display minimal 1024 x 768 (untuk mendapatkan tampilan terbaik). | resolusi display minimal 1024 x 768 (untuk mendapatkan tampilan terbaik). |
| 4.   | Ether net card | Ya  | Ya  |
| 5.   | Keyb oard      | Ya  | Ya  |
| 6.   | Mous e         | Ya  | Ya  |
| 7.   | OS             | Windows Seven   | Windows XP SP3  |

## 5. KESIMPULAN DAN SARAN

### 5.1. Kesimpulan

Pengembangan aplikasi *remote* spesifikasi *desktop* berbasis *client server* bertujuan untuk mengefisienkan aplikasi yang dapat digunakan untuk beberapa aplikasi. Dari penelitian dan tulisan yang telah penulis uraikan, maka dapat ditarik kesimpulan sebagai berikut :

1. Aplikasi ini dibuat untuk menampilkan data atau spesifikasi komputer. *Server* dapat *remote* spesifikasi komputer *client* setelah keduanya saling terhubung.
2. Data memori atau spesifikasi komputer yang ditampilkan antara lain adalah sistem operasi, MAC, resolusi layar, jenis komputer, nama *user*, kapasitas memori, *directory windows* dan sistem32, kecepatan CPU, *BrandID* CPU, serta *vendor* CPU.
3. Aktivitas *remote* yang dapat dilakukan oleh *server* adalah *remote* terhadap spesifikasi komputer, *remote control* dan *remote desktop* terhadap komputer *client*.

### 5.2. Saran

Aplikasi ini tentu saja masih belum sempurna. Masih banyak hal yang dapat dilakukan untuk mengembangkan aplikasi ini agar menjadi lebih baik lagi, antara lain :

1. Mengembangkan aplikasi ini untuk wilayah atau cakupan yang lebih besar dan luas serta dapat digunakan untuk jaringan *wireless*.

2. Pada aplikasi ini digunakan 2 *port* untuk *remote* spesifikasi dan *remote monitoring*, diharapkan pada pengembangan berikutnya *port* yang digunakan hanya satu untuk keefisiensi kerja jaringan. Selain itu dengan *remote login* agar otomatisasi *client* yang baru terhubung langsung terdeteksi..
3. Spesifikasi komputer yang ditampilkan masih terbatas pada coding, sehingga dapat dikembangkan spesifikasi komputer *client* sesuai yang diinginkan dan dibutuhkan.
4. Perlunya dikembangkan aplikasi *server* yang mampu mengakses banyak aplikasi *client* sekaligus pada satu waktu.
5. Dengan menggunakan aplikasi ini tidak mempengaruhi pengurangan biaya dalam *maintenance* jaringan.
6. Perlunya dikembangkan aplikasi sejenis yang berjalan antar sistem operasi.

## 6. DAFTAR PUSTAKA

- [1]Sofana, Iwan. 2010. CISCO CCNA & JARINGAN KOMPUTER, Informatika. Bandung.
- [2]Sutanta, edhy. 2004. Komunikasi Data & Jaringan Komputer, Graha Ilmu Bandung.
- [3]Fiade, Andrew. 2010. bab 5 komponen indy.
- [4] Wahana Komputer. 2002. Pemrograman Borland Delphi 7.0. Yogyakarta: Andi Offset
- [5] Dharwiyanti, Sri. 2003. “*Pengantar Unified Modelling Language (UML)*”, Ilmu Komputer ;
- [6]Ketut, Dharmayuda. 2007. *Program Aplikasi Client Server*. Bandung : Penerbit Informatika
- [7] Kendal & Kendal. 2005. Analisis dan Perancangan sistem. Jakarta.
- [8] Pujianto S.kom. 2007. 50 trik pemrograman delphi 8.0. Jakarta : Elek Media Komputindo.
- [9] Komunitas Delphi. Get the total and available memory. [Online] Tersedia : [http://www.delphitricks.com/source-code/systeminfo/get the total and available mem ory.html](http://www.delphitricks.com/source-code/systeminfo/get%20the%20total%20and%20available%20memory.html)
- [10]Faesal, Andri. 2009. Mengenal Borland Delphi 7.0. [Online] Tersedia : <http://andrisfaesal.blogspot.com/2009/02/mengenal-borland-delphi-70.html>