

**PERANCANGAN *INTERNET PROTOCOL PRIVATE AUTOMATIC BRANCH EXCHANGE* DENGAN BRIKER IPPBX VERSI 1.0.3**

Joni Fat, ST. ME.

ABSTRAK

Internet Protocol Private Branch Exchange (IPPBX) membawa kemampuan dunia komunikasi telefoni ke jaringan Internet Protocol (IP) sehingga memungkinkan semakin banyak layanan komunikasi yang dapat berjalan pada jaringan IP. Layanan komunikasi yang dimungkinkan adalah berupa Interactive Voice Response (IVR), Computer Telephony Integration (CTI), Unified Messaging System (UMS), fax server, call recording system, billing system, dan lain-lain. Pada era ini, IPPBX telah demikian umum sehingga muncul banyak penyedia jasa layanan yang menawarkan fasilitas ini secara gratis, seperti skype, voiprakyat, asterisk, briker, dan lain-lain.

Penelitian ini merupakan suatu eksperimen laboratorium dengan menitikberatkan pada perancangan sistem IPPBX dengan Briker IPPBX versi 1.0.3 sebagai server dan X-Lite versi 3.0 sebagai user-interface. Perancangan akan dilakukan secara bertahap dari mulai instalasi server, konfigurasi server mau pun user-interface hingga verifikasi sistem secara keseluruhan bahwa sistem dapat bekerja sesuai dengan fungsi yang diharapkan.

Penelitian ini menghasilkan kesimpulan bahwa Briker IPPBX termasuk perangkat lunak yang relatif mudah instalasi mau pun pemeliharaannya. Penelitian ini juga memperlihatkan bahwa konfigurasi fungsi-fungsi Briker juga dapat dilakukan dengan cara-cara yang relatif sederhana.

Kata kunci: *Briker, komunikasi, IP, IPPBX, IVR, telefoni, X-Lite*

I. PENDAHULUAN

Private Automatic Branch Exchange (PABX) merupakan sebuah perangkat penyambung komunikasi telepon yang terletak pada sisi pelanggan. PABX biasanya digunakan di gedung-gedung perkantoran yang memerlukan percabangan sambungan telepon sehingga komunikasi antar-bagian dalam perkantoran tersebut dapat dilakukan secara internal tanpa perlu menggunakan *line* telepon eksternal. Tetapi, secara umum, PABX juga terhubung ke penyedia layanan telekomunikasi publik. Keterhubungan dengan layanan telekomunikasi publik ini untuk memudahkan pihak internal perusahaan menghubungi pihak eksternal yang berada di luar jaringan PABX.

Namun, sejalan dengan perkembangan Internet saat ini, PABX juga telah memanfaatkan teknologi Internet yaitu melalui teknologi berbasis *Internet Protocol* (IP). Teknologi ini dikenal dengan nama *Internet Protocol Private Branch Exchange* (IPPBX). IPPBX pada dasarnya merupakan perangkat *switching* telepon dan data berbasis IP yang mengendalikan ekstensi telepon analog mau pun telepon IP. Ada pun fungsi-fungsi yang dapat dilakukan oleh IPPBX, antara lain: penyambungan, pengendalian dan pemutusan hubungan telepon, translasi protokol komunikasi, translasi media komunikasi, pengendalian perangkat IP telefoni seperti VoIP Gateway.

Pada dasarnya, IPPBX membawa kemampuan dunia komunikasi telefoni ke jaringan IP sehingga memungkinkan semakin banyak layanan komunikasi yang dapat berjalan pada jaringan IP. Layanan komunikasi yang dimungkinkan adalah berupa *Interactive Voice Response (IVR)*, *Computer Telephony Integration (CTI)*, *Unified Messaging System (UMS)*, *fax server*, *call recording system*, *billing system*, dan lain-lain. Dengan layanan-layanan ini, IPPBX dapat disetarakan dengan PABX komersial *high-end* produksi Panasonic, Alcatel, NEC, LG-Nortel, Siemens, dan lainnya.

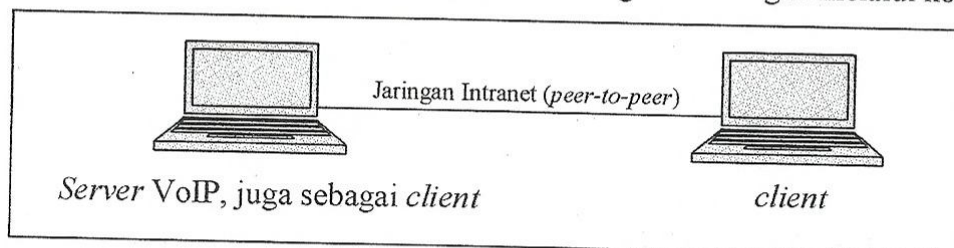
II. TINJAUAN PUSTAKA

II.1. *Voice over Internet Protocol (VoIP)*

Voice over Internet Protocol (VoIP) merupakan teknologi yang memungkinkan percakapan suara jarak jauh melalui media Internet. Jadi, secara ringkas VoIP adalah suara yang disalurkan melalui *Internet Protocol (IP)*. Sejalan dengan perkembangan teknologi, VoIP saat ini tidak hanya untuk menyalurkan suara, tetapi juga *video* mau pun data dalam bentuk paket (*packet switch*) melalui jaringan IP.

Secara prinsip, bentuk yang paling sederhana dari jaringan VoIP adalah hubungan antara dua buah komputer melalui media Intranet (*peer-to-peer*). Dengan demikian, syarat dasar untuk melakukan koneksi VoIP adalah:

- Dua buah komputer (dilengkapi dengan *Network Interface Card*, mikrofon dan *speaker*) yang terhubung melalui jaringan *peer-to-peer*.
- Perangkat lunak khusus pendukung untuk saling berhubungan melalui koneksi VoIP.



Gambar 1. Diagram Interkoneksi VoIP dalam Bentuk Paling Sederhana

Kemampuan koneksi VoIP tergantung pada banyak faktor. Secara garis besar, ada tiga faktor yang perlu diperhatikan, yaitu: manajemen jaringan, kemampuan *server VoIP*, dan aplikasi yang menjadi *interface* bagi pengguna dalam melakukan koneksi VoIP.

Dalam hal ini, aspek manajemen jaringan diasumsikan telah didukung oleh jaringan yang berkualitas. Jaringan berkualitas ini ditentukan oleh berbagai macam faktor, yang menjadi poin dalam perancangan koneksi VoIP adalah *bandwidth*, dengan anggapan bahwa gangguan yang mungkin terjadi terhadap jaringan seperti *noise*, *interference*, dan lain-lain telah tertangani dengan baik. Walau demikian, ukuran *bandwidth* yang diperlukan tidak memiliki patokan yang pasti. Ini tergantung pada kapasitas *concurrent calls* yang ingin dicapai dalam perancangan. *Concurrent calls* sendiri tergantung pada jenis *codec*, yang merupakan algoritma untuk mengkonversi gelombang suara analog menjadi serangkaian informasi *digital*, yang digunakan. Kebutuhan *bandwidth* berbagai macam *codec* dapat diperhatikan pada Tabel 1 di bawah ini.

Tabel 1. Jenis *Codec* dan Besar *Bandwidth* yang Diperlukan¹

Coding algorithm		Bandwidth
G.711	PCM	64kbps
G.723.1	ACELP	5.6kbps
	MP-MLQ	6.4kbps
G.726	ADPCM	32kbps
G.728	LD-CELP	16kbps
G.729(A)	CS-ACELP	8kbps

Dari Tabel 1, misalkan koneksi VoIP yang akan dirancang mampu menangani 25 *concurrent calls* dengan menggunakan *codec* G.711, maka kebutuhan *bandwidth* adalah sebesar $(64 \text{ Kbps} + 16 \text{ Kbps}) * 25 = 2 \text{ Mbps}$. Kebutuhan *bandwidth* *codec* G.711 adalah sebesar 64 Kbps ditambah dengan 16 Kbps yang merupakan periode *sampling* (periode 0.125 ms, ditransmisikan dengan *datagram* 20 ms). Dengan demikian, berdasarkan kemungkinan besarnya *concurrent calls* dapat diperkirakan lebar *bandwidth* yang diperlukan. Sebaliknya, aspek manajemen jaringan berkaitan dengan alokasi *bandwidth* untuk berbagai keperluan aplikasi dalam jaringan. Ini diperlukan agar koneksi VoIP tidak menyebabkan jaringan *down* sehingga aplikasi lain menjadi terganggu.

Aspek kedua yaitu kemampuan *server* VoIP. Beberapa hal yang perlu diperhatikan dari kemampuan *Server* VoIP adalah jumlah nomor ekstensi yang didukung; jumlah *concurrent calls* yang dapat ditangani; *features* khusus seperti *Interactive Voice Response* (IVR), *ring group*, *call forward*, dan lain-lain; protokol VoIP yang didukung, seperti SIP, H.323, IAX2, dan ZAP; dukungan terhadap berbagai macam *voice codec*, seperti *ulaw*, *alaw*, *gsm*, *g723* dan *g729*; dukungan terhadap berbagai macam *video codec*, seperti *h264*, *h263p*, *h263*, dan *h261*; serta kemudahan dalam melakukan administrasi.

Aspek ketiga adalah aplikasi yang digunakan sebagai *interface*. Aplikasi ini juga terkait dengan kemampuan *server* VoIP yang meliputi dukungan terhadap protokol yang digunakan dan juga fitur-fitur lainnya. Aplikasi yang mumpuni dapat memanfaatkan semua sumber daya yang disediakan oleh *server*.

II.2. *Private Branch Exchange* (PBX)

Private Branch Exchange (PBX) merupakan sentra telepon atau jaringan *switch* telepon yang berada pada sisi pemakai, misalnya untuk lingkungan perkantoran. Istilah PBX juga dipakai untuk merujuk istilah *Private Automatic Branch Exchange* (PABX) dan juga *Electronic PABX* (EPABX). Jadi, PBX merupakan jaringan telepon yang sifatnya pribadi atau internal pemakai. Ini berlainan dengan jaringan telepon publik yang dapat diakses oleh siapa saja. PBX memerlukan koneksi khusus agar dapat diakses oleh jaringan di luarnya. Koneksi khusus yang menghubungkan jaringan internal dengan jaringan umum atau pun *Public Switch Telephone Network* (PSTN) dinamakan *trunk*. Tujuan utama pengadaan PBX adalah untuk menurunkan tarif telepon antar-pemakai dalam jaringan atau kelompok yang sama.



Secara prinsip, sistem PBX terdiri dari beberapa komponen utama, yaitu:

- Jaringan *switching* internal.
- Pemroses data dan pengontrol, biasanya berupa mikrokontroler, mikrokomputer atau komputer.
- Unit telepon atau terminal yang dinamakan *lines*.
- *Trunk* apa bila diperlukan hubungan dengan jaringan eksternal lain, seperti PSTN atau PBX lainnya.
- Konsol untuk membantu melakukan administrasi *server* mau pun jaringan.

Dengan dukungan komponen-komponen tersebut, secara fungsional, PBX dapat melakukan empat proses utama, yaitu: mengadakan hubungan (jaringan) antara unit telepon atau terminal di antara dua pemakai (memetakan nomor yang di-*dial* ke unit fisik dan memastikan unit tersebut tidak sedang dipakai), menjaga koneksi sepanjang diperlukan, memutuskan koneksi atas permintaan pemakai dan melakukan fungsi pencatatan untuk keperluan pembukuan.

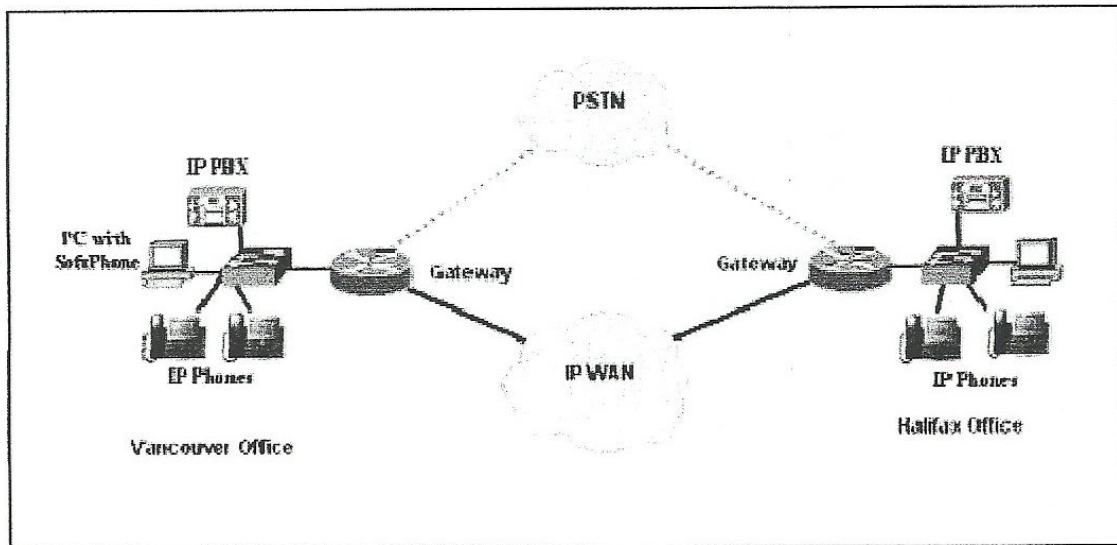
Dalam perkembangannya, PBX memanfaatkan VoIP sehingga dikenal dengan nama VoIP PBX atau *Internet Protocol (IP) PBX* atau *Internet Protocol Branch Exchange (IPBX)*. IP PBX ini memanfaatkan protokol Internet dalam melakukan panggilan.

II.3. IPPBX

Internet Protocol Private Branch Exchange (IPPBX) merupakan sistem telepon yang didesain untuk mengirimkan suara melalui jaringan data mau pun melalui PSTN. Seperti telah disebutkan pada akhir subbab II.2, IPPBX memanfaatkan teknologi protokol Internet (IP). IPPBX memiliki semua fitur-fitur dari sebuah PABX dan juga keunggulan lain seperti IPPBX dapat berupa *hardware* mau pun hanya sekedar *software*. Untuk IPPBX dalam bentuk *software*, aplikasi ini harus didukung oleh pemroses atau komputer dalam penerapannya.

Ada pun fungsi-fungsi yang dapat dilakukan oleh IP PBX adalah seperti:

- Konferensi antar-pemakai
- *Interactive Voice Response (IVR)*
- *Text-to-Speech (TTS)*
- *Automatic Speech Recognition (ASR)*
- *Call forward*
- *Ring Group*
- Interkoneksi dengan jaringan PSTN baik secara analog mau pun digital
- Dukungan terhadap VoIP, dan lain-lain.

Gambar 2. Diagram Interkoneksi VoIP²

Pada Gambar 2 di atas, memperlihatkan diagram interkoneksi VoIP, IP PBX melakukan fungsi pengaturan dalam internal terhadap *personal computer* (PC) mau pun *IP phones*. Untuk melakukan koneksi ke luar jaringan internal, diperlukan *gateway*. Koneksi *gateway* juga dapat dilakukan dalam beberapa cara, antara lain melalui *Public Switched Telephone Network* (PSTN), *Wide Area Network* (WAN), dan lain-lain. Pemilihan ini dapat diatur pada IP PBX berdasarkan pertimbangan biaya, kualitas mau pun ketersediaan jaringan. Integrasi VoIP ke PSTN dimungkinkan dengan adanya standar H.323, *Session Initiation Protocol* (SIP), protokol *Inter-Asterisk eXchange* versi 2 (IAX2), dan lain-lain.

III. HASIL PERANCANGAN

III.1. Perancangan Jaringan Komputer

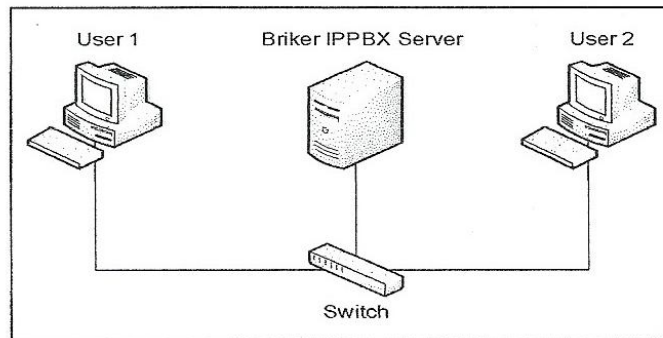
Jaringan komputer yang dibangun di sini adalah jaringan lokal. Jaringan ini terdiri tiga buah komputer, satu buah *switch* dan jaringan dari Fakultas Teknik (mencakup jaringan Internet). Spesifikasi komputer yang dipakai adalah prosesor Intel Pentium IV 3.2 GHz, memori *random access* (RAM) 512 MB, dan sistem operasi Microsoft XP SP 2.

Untuk keperluan penelitian ini, dibangun dua macam jaringan komputer. Jaringan pertama untuk menguji Briker IPPBX sebagai *server* IP PBX yang berdiri sendiri dan berfungsi sebagai PABX Internal. Jaringan kedua untuk menguji Briker IPPBX dalam kaitannya dengan *server* pihak lain (*trunking*), yang dalam hal ini adalah *server* VoIP Rakyat (www.voiprakyat.or.id).

Jaringan pertama terdiri dari tiga buah komputer, dan satu buah *switch*. Jaringan tersebut diperlihatkan pada Gambar 3. Komputer yang difungsikan sebagai *server* Briker IPPBX memiliki alamat IP 192.168.10.30, *gateway* 192.168.10.1. Alamat IP *user 1* adalah 192.168.10.27, sedangkan alamat IP *user 2* adalah 192.168.10.28. Koneksi antar-komputer dengan *switch* menggunakan kabel *Unshielded Twisted Pair* (UTP)—perhatikan Gambar 5—dengan konektor RJ-45 dan *Network Interface Card* (NIC) merk Realtek RTL8139 PCI slot. Kabel UTP dirangkai untuk keperluan *crossover*. Pada komputer masing-masing *user* telah terhubung segala

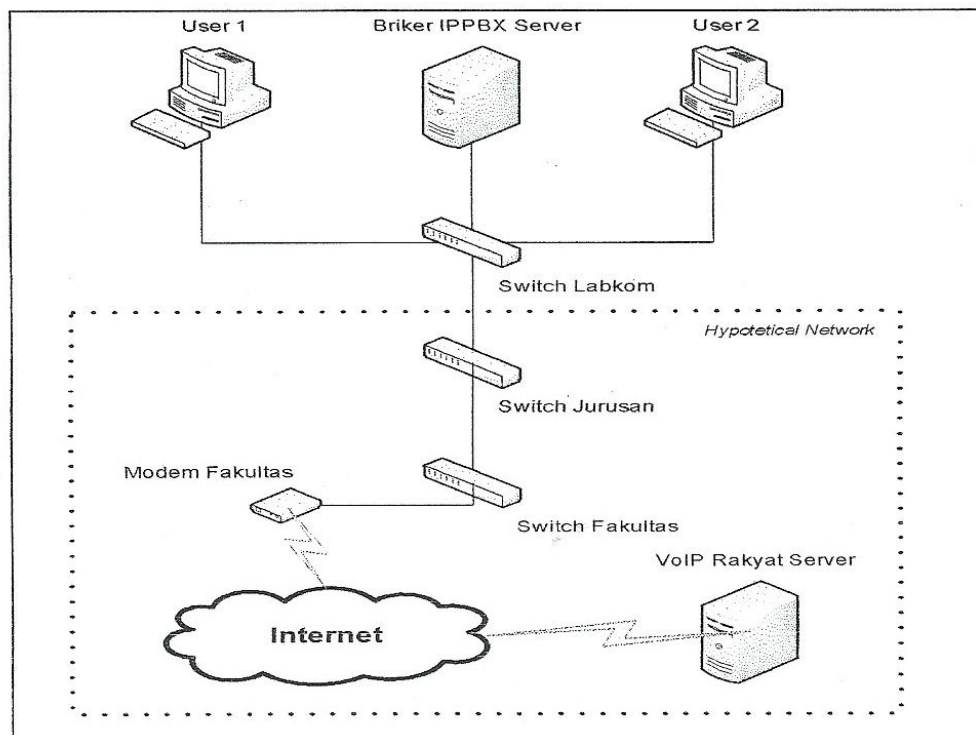
peripheral yang diperlukan untuk melakukan komunikasi, misalnya *microphone*, *speaker*, *webcam*, *video card onboard* dan juga *sound card* (merk Realtek AC' 97).

Jaringan kedua memiliki konfigurasi yang serupa dengan jaringan pertama ditambah keterhubungan dengan jaringan Internet. Koneksi ke Internet memanfaatkan jaringan yang telah disediakan oleh Fakultas Teknik Universitas Tarumanagara. Kelemahan dari koneksi ini adalah kualitas koneksi Internet menjadi tidak terkontrol karena di luar lingkup jaringan yang dibangun oleh Peneliti. Gambar jaringan dapat dilihat pada Gambar 4.



Gambar 3. Topologi Jaringan Pertama

Gambar 4 memperlihatkan bagian yang berada dalam persegi panjang putus-putus. Ini adalah gambar jaringan yang bukan sebenarnya. Peneliti hanya menyimpulkan berdasarkan kondisi jaringan yang ada. Jaringan tersebut juga pada dasarnya di luar lingkup penelitian.



Gambar 4. Topologi Jaringan Kedua



III.2. Instalasi Briker IPPBX Versi 1.0.3

Setelah jaringan terpasang, selanjutnya dilakukan instalasi perangkat lunak Briker IPPBX versi 1.0.3 pada komputer yang akan dimanfaatkan sebagai *server* VoIP. Spesifikasi komputer yang digunakan sebagai *server* VoIP ini adalah sebagai berikut:

- Monitor LCD 14 inci (setelah proses instalasi selesai, monitor tidak lagi diperlukan karena administrasi dan *maintenance* dapat dilakukan via *remote*).
- Prosesor Intel Pentium IV 3.20 GHz.
- RAM 512 MB.
- VCD ROM LG 52x.
- *Harddisk* 40 GB.
- Disertai aksesoris komputer lain seperti *keyboard* dan *mouse*.

Kondisi komputer yang akan dijadikan sebagai *server* VoIP ini adalah dalam kondisi tidak terformat, jadi belum memiliki sistem operasi. Perangkat lunak Briker diunduh dari situs www.briker.org. Hasil unduhan harus di-*burn* ke *compact disc* (CD). CD ini digunakan sebagai *installer* Briker.

Setelah proses instalasi selesai, harus dilakukan konfigurasi terhadap *server* terutama yang menyangkut identitas dan alamat *Internet Protocol* (IP). Apa bila identitas dan alamat IP tidak diganti, maka *server* akan memakai identitas dan alamat *default*-nya. *Default* identitas adalah:

Username : *support*
Password : *Briker*

Sedangkan *default* alamat IP adalah:

IP address : 192.168.2.2
Subnet mask : 255.255.255.0

Untuk alasan keamanan, peneliti mengganti *password* identitas. Alamat IP juga diganti dengan penyesuaian terhadap alamat di jaringan yang peneliti gunakan. Penggantian *password* dilakukan dengan menjalankan perintah **#passwd** pada *console*. Setelahnya, sistem akan meminta *password* lama untuk identifikasi dan *password* baru yang akan digunakan lebih lanjut.

Proses penggantian alamat IP sedikit berbeda. Alamat IP disimpan oleh sistem pada *file interfaces* yang berada di *folder /etc/network/*. *File* ini di-*edit* dengan menggunakan *editor* yang disediakan oleh sistem operasi. Di sini, peneliti menggunakan *editor mcedit*. Untuk dapat menyimpan *file* hasil meng-*edit*, terlebih dahulu *user* diganti *level*-nya menjadi *super user*. Untuk mengubah *level user*, digunakan perintah **sudo -s**. *Level super user* atau *root* memungkinkan *user* lebih leluasa memanipulasi *file* tersebut.

Setelah mengganti *level*, pada *console* diketik perintah **mcedit /etc/network/interfaces**. *Editor* yang berisikan data *file interfaces* akan tampil pada layar monitor. Ada pun data-data pada *file interfaces* yang diubah adalah:

- *Address*: 192.168.2.2 menjadi 192.168.10.30
- *Network*: 192.168.2.0 menjadi 192.168.10.0
- *Broadcast*: 192.168.2.255 menjadi 192.168.10.255
- *Gateway*: 192.168.2.1 menjadi 192.168.10.1



File tersebut disimpan dengan menekan tombol **F2**. Pada *console*, diketikkan kembali perintah **/etc/init.d/networking restart**. Ini berfungsi untuk mengaktifkan perubahan yang telah dilakukan sebelumnya.

III.3. Penambahan Ekstensi

Ekstensi pada sebuah IPPBX merupakan identitas bagi *user*-nya. Ini serupa dengan nomor telepon pada jaringan PSTN. Ekstensi ini bersifat lebih lokal dan hanya dikenali dalam lingkungan IPPBX yang bersangkutan. Hubungan antar-*user* dalam jaringan IPPBX adalah melalui ekstensi ini. Demikian juga apabila ada *user* di luar jaringan yang menghubungi jaringan IPPBX (bila jaringan IPPBX diatur juga agar dapat menerima panggilan di luar jaringannya), maka ekstensi adalah juga pengenalan utama bagi masing-masing *user*.

Briker dapat menampung ekstensi hingga 1000 nomor dan mampu menangani maksimum 240 *concurrent call* untuk jaringan yang mendukung. Ini tentu saja memperlihatkan kemampuan Briker yang sangat besar karena dengan jumlah ekstensi yang sedemikian banyak. Penambahan ekstensi dalam Briker dilakukan dengan cara yang relatif sederhana. Kolom yang perlu diisi dalam pembuatan ekstensi ini diantaranya adalah nama, nomor ekstensi, jenis protokol yang

digunakan, serta kolom *secret* yang berisi *password* yang akan digunakan untuk proses otentikasi *user* pada saat registrasi ekstensi melalui *user agent*.

III.4. Pengaturan *Interactive Voice Response* (IVR)

IVR merupakan fungsi Briker yang melakukan tugas penjawab secara otomatis saat ada panggilan masuk. IVR memungkinkan sistem untuk melakukan pendeteksian terhadap suara mau pun masukan dari *keypad*.

Di sistem Briker ini, IVR melakukan fungsi yang serupa dengan *digital receptionist*. Ini berarti IVR mampu mendeteksi masukan dari *keypad* oleh *user*. IVR memungkinkan antrian *user* secara otomatis, sehingga *user* tidak merasa terabaikan saat *line* telepon sedang padat.

Untuk dapat memanfaatkan sistem IVR di Briker, pertama-tama Peneliti melakukan *uploading* terhadap *file* suara yang akan digunakan sebagai *digital receptionist*. *Upload* dilakukan dengan melakukan *login* ke *server* IPPBX dengan *user* administrator. Selanjutnya, Peneliti memilih *menu* *IPPBX Administration > System Recordings*. Sebelum *upload* dapat dilakukan, Peneliti telah menyiapkan rekaman suara yang disimpan dengan format *.wav*. Format *file* *.wav* ini perlu memenuhi kriteria sebagai berikut: *encode* 16 bit, 8000 Hz agar dapat digunakan sebagai *file* IVR oleh Briker.

Setelah proses *upload* selesai dilakukan, untuk menguji validitas *file* tersebut, dilakukan dengan menghubungi *server* via telepon VoIP. Dari telepon VoIP, Peneliti melakukan dial nomor *99. Bila *file* ter-*upload* dengan sempurna, maka hasil suara rekaman terdengar dengan jelas. Pengkonfigurasi IVR selanjutnya dilakukan pada *menu* IVR yang terdapat pada *IPPBX Administration*.

III.5. Pengaturan *Ring Group*

Ring group berarti satu kelompok yang sama. Di Briker, sekelompok *user* dapat digabungkan untuk menangani satu keperluan yang sama. Seperti contoh untuk departemen *customer service*. Misalkan di suatu perusahaan memiliki *customer service* sebanyak tiga orang.



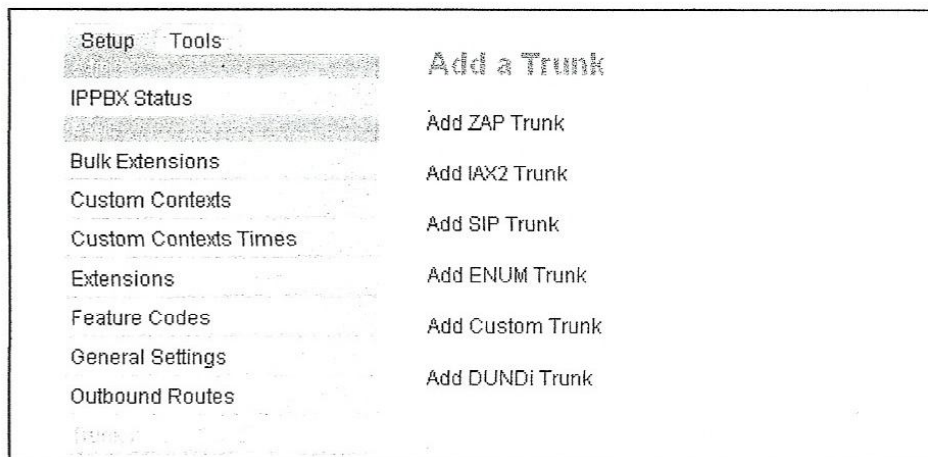
Agar ketiga orang ini dapat menangani pertanyaan pelanggan dengan lebih efektif, maka sebaiknya ketiga orang tersebut dapat menerima panggilan mana pun yang ditujukan kepada departemen itu.

Seorang pelanggan yang menghubungi perusahaan tersebut, akan diterima oleh *customer service* yang sedang tidak sibuk. Jadi, pelanggan tidak perlu mengantri hanya kepada satu *customer service* tertentu. Ini tentu saja akan memperbaiki kinerja departemen bersangkutan.

Pada penelitian ini, Peneliti mendefinisikan satu kelompok yang terdiri dari tiga orang yang akan bertindak sebagai *customer service*. Bila pelanggan men-*dial* ke nomor 810, maka Briker akan men-*forward* ke grup ini dan akan diangkat oleh *customer service* mana pun yang sedang tidak menangani panggilan. Ekstensi ketiga *customer service* ini secara masing-masing adalah di nomor 811, 812 dan 813. Melalui fasilitas *ring group*, ketiganya dikelompokkan ke dalam nomor 810. Selanjutnya, apa bila pelanggan men-*dial* ke nomor 810 dan dalam kasus ini ketiga *customer service* sedang menangani panggilan, maka pelanggan akan diarahkan untuk mendengarkan pesan yang telah diatur. Dalam kasus ini adalah pesan yang telah direkam dalam format .wav (SelamatDatang.wav) yang akan menyambut pelanggan dengan ucapan selamat datang dan kesabarannya untuk menunggu.

III.6. Konfigurasi Trunks

Trunks yang diperlukan untuk menghubungkan dengan *server VoIP Rakyat* adalah *SIP Trunk*. Penambahan *trunk* ini dilakukan dengan cara memilih *menu Trunks* pada *menu IPPBX Administration*. Pemilihan *menu Trunks* akan memunculkan pilihan seperti yang terlihat pada Gambar 5. Pada gambar ini terlihat berbagai jenis *trunks* yang didukung oleh Briker.



Gambar 5. Jenis-Jenis *Trunks* yang Didukung oleh Briker

Selanjutnya Peneliti memilih pilihan *Add SIP Trunk*. Pilihan ini akan menampilkan serangkaian kolom yang perlu diisi. Peneliti mengisi sesuai dengan data yang diperoleh dari *server VoIP Rakyat*. Kolom pertama adalah *Trunk Name*, kolom ini diisi dengan nama *trunk* yang dapat dipilih dengan nama yang representatif. Kolom kedua adalah *PEER Details*, diisi dengan nomor akun yang diperoleh dari *VoIP Rakyat*. Kolom *fromuser*, *authuser*, *username* diisi dengan nomor akun yaitu 104062, kolom *secret* diisi dengan *password* yang diperoleh dari *VoIP*



Rakyat, yaitu MRM8B3. Selanjutnya adalah kolom *host* dan *fromdomain*, diisi dengan alamat *server* VoIP Rakyat, yaitu *voiprakyat.or.id*. *Register String* diisi dengan format *<nomor akun>:xxxxx@<nama trunk>/<nomor akun>*. *<Nomor akun>* adalah nomor akun yang diperoleh dari VoIP Rakyat, yaitu 104062, sedangkan *<nama trunk>* adalah nama *trunk* yang didefinisikan dalam penambahan *trunk*, yang pada penelitian ini diberi nama *voiprakyat-link*.

III.7. Konfigurasi Outbound Routes

Outbound routes merupakan *menu* yang mendefinisikan cara dan bagaimana panggilan ke luar *server* Briker lokal dilakukan. *Outbound routes* dikonfigurasi pada *menu IPPBX Administration* dengan memilih *menu Outbound Routes*, selanjutnya memilih *Add Route*. Kolom *Route Name* diisi dengan nama yang diinginkan, dalam hal ini, Peneliti menuliskan nama *VoIP Rakyat*. Kolom *Dial Patterns* diisi dengan pola *dial* yang diinginkan bila ingin menghubungi *server* VoIP Rakyat. Di penelitian ini, Peneliti menggunakan pola "9|.", yang berarti untuk menghubungi nomor akun VoIP Rakyat lain, *user* Briker harus mendahului dengan menekan tombol 9. Contoh untuk menghubungi nomor VoIP Rakyat 10390, maka *user* Briker harus menekan nomor "910390". Selanjutnya, mengisi kolom *Trunk Sequence*. Kolom ini diisi dengan nama *trunk* yang telah dibuat pada subbab III.6.

Setelah kolom-kolom tersebut diisi dengan lengkap, maka tombol *Submit Changes* ditekan, *server* Briker akan melakukan *update* terhadap informasi ini. Untuk mengaktifkan konfigurasi ini, ada satu tahap lagi yang diperlukan, yaitu menekan tombol *Apply Configuration Changes*.

III.8. Konfigurasi Inbound Routes

Inbound routes digunakan untuk mendefinisikan jalur bila ada panggilan dari luar *server* Briker ke *server* tempat *user* berada. Pada penelitian ini, luar *server* adalah *server* VoIP Rakyat. Bila ada panggilan dari luar, sistem akan mengecek kesesuaiannya dengan konfigurasi *Inbound Routes* untuk menentukan tindakan yang perlu dilakukan. *Inbound routes* dikonfigurasi dengan memilih *menu IPPBX Administration*, kemudian memilih *menu Inbound Routes* dan *Add Incoming Route*. Kolom *Description* diisi dengan keterangan tentang *inbound routes*, yang dalam penelitian ini diisi dengan keterangan "voiprakyat". Kolom *DID Number* diisi dengan nomor akun VoIP Rakyat, yaitu 104062. Penelitian ini mengarahkan panggilan dari luar ini menuju *Customer Service* yang sudah dikelompokkan dalam satu kelompok atau *ring group*.

III.9. Instalasi Aplikasi Soft Phone

Dalam melakukan komunikasi VoIP, setelah *server* di-set up, selanjutnya adalah melakukan instalasi *soft phone* di masing-masing komputer *user*. *Soft phone* ini merupakan *interface* antara *user* dengan *server* VoIP. *Soft phone* yang dipakai dalam penelitian ini adalah *X-Lite 3.0*.

Soft phone ini dikonfigurasi sedemikian rupa sehingga terhubung dengan *server* Briker. Konfigurasi dilakukan dengan cara sebagai berikut:

1. Memilih *menu SIP account settings*.
2. Memilih *menu Add*



3. Mengisi kolom yang tersedia dengan nomor akun *Briker* 800 dan *password* "labkom". Nomor akun dan *password* ini merupakan ekstensi dan verifikasi ekstensi yang telah dibuat di *server Briker*.

IV. PENUTUP

Setelah perancangan dan uji coba terhadap *server Briker* ini dilakukan baik dalam melakukan panggilan di lingkungan jaringan lokal mau pun terhadap jaringan di luar *server*, yaitu www.voiprakyat.or.id, maka berikut adalah kesimpulan yang dapat Peneliti ambil:

1. Menilik cara instalasi *server Briker*, dapat dikatakan bahwa instalasi *server* relatif mudah dilakukan. Ini dikarenakan proses instalasi pada dasarnya berjalan secara otomatis. *Software Briker* juga menyertakan sistem operasi Linux dan diinstalasi langsung bersama dengan *server*-nya. Ini adalah salah satu alasan tambahan kenapa Peneliti mengatakan bahwa cara instalasi *server* ini relatif mudah. Konfigurasi alamat jaringan untuk *server* juga relatif mudah dilakukan karena hanya terdiri dari dua tahap, yaitu tahap pengaturan dan tahap *restart*.
2. Pengaturan fitur *inbound* mau pun *outbound routes* dapat dilakukan dengan relatif mudah karena cukup dilakukan konfigurasi secara *software* saja, yaitu melalui pengesetan di halaman administrasi *Briker*. Ini dilakukan tanpa melibatkan perangkat keras apa pun sebagai tambahan. Jadi, dengan demikian Peneliti menyimpulkan bahwa fitur tersebut dapat dilakukan dengan relatif mudah.
3. *User management* pada *server* juga dapat dilakukan dengan cara yang relatif mudah, karena semua hal yang diperlukan telah disusun di halaman administrasi *Briker*.

V. DAFTAR PUSTAKA

- [1] A. Raharja dan A. Wardhana, *Administration Guide: Briker*, Agustus 2008.
- [2] W. C. Hardy, *VoIP Service Quality: Measuring and Evaluating Packet-Switched Voice*, New York: McGraw-Hill, 2003.
- [3] *VoIP Bandwidth*, Tersedia di <http://www.erlang.com/bandwidth.html> (10 Agustus 2009).
- [4] A. B. Carlson, P. B. Crilly, and J. C. Rutledge, *Communication Systems: An Introduction to Signals and Noise in Electrical Communication*, 4th ec. New York: McGraw-Hill, 2002.
- [5] B. Douskalis, *Putting VoIP to Work: Softswitch Network Design and Testing*, New Jersey: Prentice Hall, 2002.
- [6] Davidson, Jonathan, J. Peters, *Voice over IP Fundamentals*, Indianapolis: Cisco Press, 2000.