BAB II
TINJAUAN PUSTAKA

2.1 Keamanan Data

Keamanan data adalah ilmu pengetahuan dan pembelajaran mengenai metode perlindungan data pada komputer dan sistem komunikasi [1]. Keamanan data meliputi beberapa aspek di antaranya privasi (kerahasiaan), *integrity* (konsisten), *authenticity* (keaslian), *availability* (ketersediaan), dan *access control* [5].

2.1.1 Privasi (Kerahasiaan)

Privasi merupakan suatu hal yang sangat penting baik bagi individu maupun lembaga atau instansi untuk berhadapan dan berinteraksi dengan individu lain atau lembaga lain [6]. Pada umumnya terdapat tiga aspek dari privasi yaitu privasi mengenai pribadi seseorang (*Privacy of a Person's Persona*), privasi tentang data seseorang (*Privacy of Data about a Person*), dan privasi atas komunikasi seseorang (*Privacy of a Person's Communication*). Penggunaan data seseorang oleh lembaga pemerintah ataupun swasta, perusahaan ataupun perseorangan tanpa seizin pemilik merupakan pelanggaran privasi seseorang [5].

2.1.2 *Integrity* (Konsisten)

Integritas data digunakan untuk menjamin bahwa data yang digunakan benar-benar asli dan dikirimkan oleh pihak yang benar. Integritas juga harus mampu menjamin bahwa data yang dikirimkan belum mengalami perubahan saat pengiriman. Untuk memastikan data tersebut benar dan dikirimkan oleh pihak yang benar, maka diperlukan metode. Metode yang sering digunakan adalah enkripsi [5].

Enkripsi merupakan metode yang digunakan untuk mengubah kata atau data menjadi sebuah kode-kode yang tidak dapat dimengerti oleh seseorang termasuk komputer. Untuk dapat mengetahui data yang sebenarnya diperlukan satu metode yaitu dekripsi. Dekripsi akan dilakukan dengan cara mengubah kode-kode yang tidak dapat dimengerti menjadi data sebenarnya [5].

2.1.3 *Authenticity* (Keaslian)

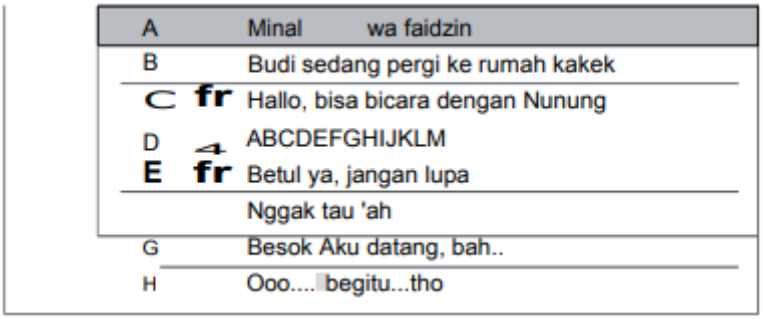
Keaslian data yang diterima oleh penerima informasi harus benar-benar terjaga. Keaslian data merupakan hal yang sangat penting, karena jika data yang diperoleh ternyata telah diubah oleh pihak yang tidak berhak maka akan sangat berbahaya. Enkripsi juga akan mampu membuktikan bahwa data yang diperoleh benar-benar berasal dari pengirim yang asli dan data yang dikirimkan juga benar-benar asli [5].

2.1.4 *Availability* (Ketersediaan)

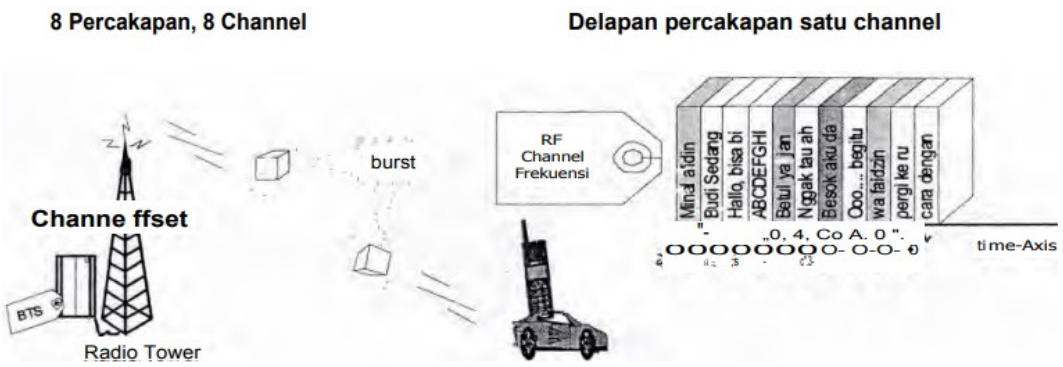
Data dan informasi yang berada dalam suatu sistem komputer tersedia dan dapat dimanfaatakan oleh orang yang berhak. Aspek *availability* atau ketersediaan berhubungan dengan ketersediaan informasi saat dibutuhkan [5].

2.2 Konsep Kanal (*Channel*) GSM

Kanal terdiri atas dua jenis yaitu kanal fisik dan kanal *logic*. Pada kanal fisik, satu *timeslot* (TS) *frame* TDMA merupakan satu kanal fisik. Setiap *carrier* RF (*Radio Frequency*) terdiri atas 8 TS (CH/*channel* 0-7). Pada kanal *logic*, kanal trafik (TCH) dapat membawa data untuk layanan komunikasi. TCH terbagi menjadi dua jenis, yaitu *full rate channel* dengan *bit rate* 13 Kbps dan *half rate channel* dengan kecepatan bit 6,5 Kbps. Kemudian, kanal control digunakan untuk proses pensinyalan (*signalling*) [7].

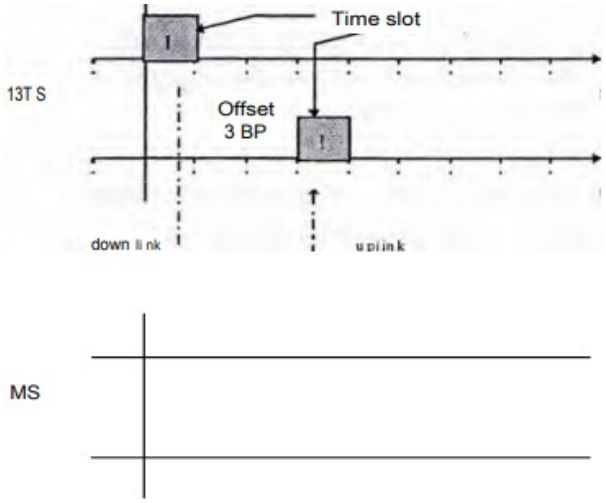


Gambar 2.1 Pembagian Komunikasi Dalam *Timeslot* [7]

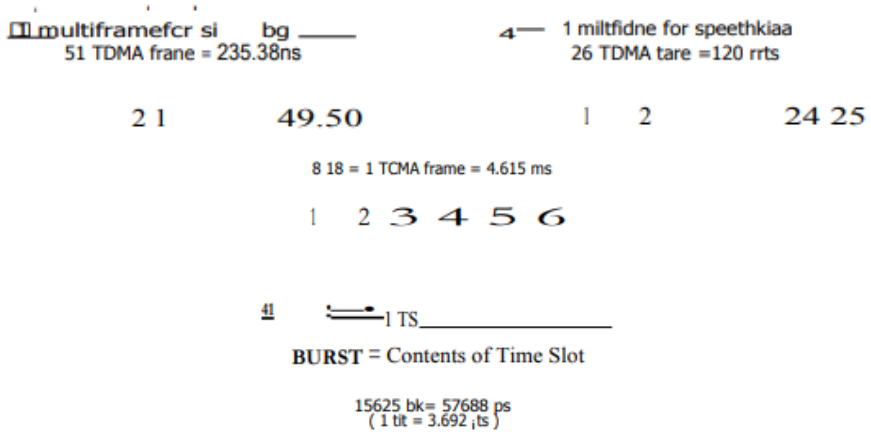


Gambar 2.2 Proses Komunikasi Antara BTS dan MS [7]

Komunikasi antara BTS dan MS melalui kanal fisik berupa *burst*. BTS dalam mengirim dan menerima sinyal secara dupleks, artinya dalam waktu yang bersamaan BTS mengirim dan menerima sinyal. Untuk data yang bersesuaian atau yang mempunyai *timeslot* yang sama, BTS memberikan konstanta *offset* selama tiga periode *burst* (3×577 s). Tujuan adanya *offset* adalah untuk mencegah sinyal *downlink* dan *uplink* diterima oleh MS bersamaan. Selain itu, Tx (*transmitter*) dan Rx (*receiver*) tidak bekerja bersamaan dan tidak bisa saling menginterferensi satu sama lain [7].



Gambar 2.3 *Offset* (Selang Waktu) Antara Tx (*Downlink*) dan Rx (*Uplink*) [7]



Gambar 2.4 Struktur *Frame* GSM [7]

ARFCN (*Absolute Radio Frequency Channel Number*) adalah kode yang menentukan sepasang *carrier* radio dan *channel* yang digunakan untuk pemancar dan penerima di *Um interface*, satu untuk sinyal *uplink* dan satu untuk sinyal *downlink*. Setiap ARFCN mempunyai *bandwidth* sebesar 200 kHz. Beda Frekuensi (ARFCN) digunakan untuk komponen *frequency based* dari GSMs *multiple access sheme* (FDMA). Bersama dengan komponen *time based* (TDMA), maka kanal fisik didefinisikan dengan memilih ARFCN dan *timeslot* tertentu [8].

2.3 TDMA (*Time Division Multiple Access*)

Sistem TDMA (*Time Division Multiple Access*) menerapkan sistem pembagian waktu untuk meningkatkan kapasitas sistem. Satu kanal pada suatu frekuensi dibagi menjadi beberapa *timeslot*, sehingga kapasitas sistem meningkat. TDMA diimplementasikan pada jaringan GSM dimana satu *band* frekuensi dibagi menjadi delapan *timeslot*. Berbeda dengan sistem FDMA (*Frequency Division Multiple Access*) dimana setiap kanal dibedakan berdasarkan pembagian frekuensi. Setiap kanal menempati satu frekuensi dengan *bandwidth* 30 kHz. Jadi, hanya satu pemakai yang dapat menggunakan kanal frekuensi tersebut setiap waktunya.

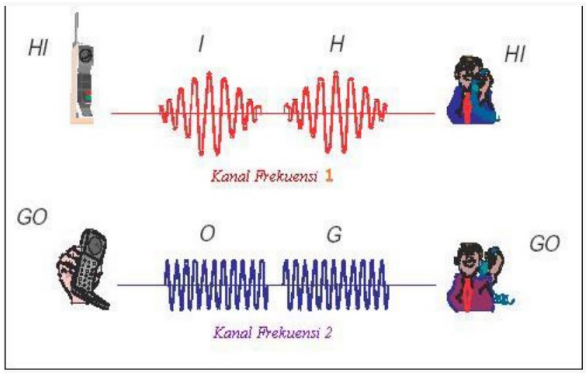
Proses kerja TDMA dengan satu *channel* dapat membawa delapan data atau komunikasi dengan cara membagi-bagi data tersebut ke dalam fragmen-fragmen yang disebut *timeslot* [7].

2.4 FDMA (*Frequency Division Multiple Access*)

Dalam FDMA (*Frequency Division Multiple Access*) frekuensi dibagi menjadi beberapa kanal frekuensi yang lebih sempit. Setiap pengguna akan mendapatkan kanal frekuensi yang berbeda untuk berkomunikasi secara bersamaan [9].

Pengalokasian frekuensi pada FDMA bersifat ekslusif, karena kanal frekuensi yang telah digunakan oleh satu pengguna tidak dapat digunakan oleh pengguna yang lain. Antar kanal dipisahkan dengan bidang frekuensi yang lebih sempit lagi (*guard band*) untuk menghindari interferensi antar kanal yang berdekatan (*adjacent channel*) [9].

Informasi bidang dasar yang dikirim, ditumpangkan pada isyarat pembawa (*carrier signal*) agar menempati alokasi frekuensi yang diberikan [9].



Gambar 2.5 Cara Kerja FDMA [9]

2.5 Alokasi Frekuensi Seluler di Indonesia

Alokasi frekuensi seluler GSM yang digunakan di Indonesia sama dengan alokasi di sebagian besar wilayah di dunia terutama Eropa yaitu *band* 900 MHz yang dikenal sebagai GSM900, *band* 1800 MHz yang dikenal sebagai GSM1800 atau DCS, dan *band* 2100 MHz [10].

Frekuensi *downlink* adalah frekuensi yang dipancarkan oleh BTS untuk berkomunikasi dengan perangkat pelanggan dan juga menghasilkan *coverage footprint* operator. Sedangkan, frekuensi *uplink* adalah frekuensi yang digunakan oleh perangkat pelanggan agar dapat terhubung ke jaringan [10].

Gambar 2.6 Alokasi Frekuensi *Band* 900 MHz [11]

Untuk *uplink*, alokasi frekuensi GSM900 dari 880 MHz sampai 895 MHz, sedangkan untuk *downlink* dari 925 MHz sampai 940 MHz pada *provider* Telkomsel. Untuk *uplink*, alokasi frekuensi GSM900 dari 895 MHz sampai 907,5 MHz, sedangkan untuk *downlink* dari 940 MHz sampai 952,5 MHz pada *provider* Indosat [11].

Untuk *provider* Telkomsel, dalam frekuensi MHz baik *uplink* maupun *downlink* mempunyai alokasi frekuensi yang berbeda, namun dengan penomoran kanal ARFCN keduanya sama, karena keduanya adalah pasangan kanal dupleks yang dipisahkan selebar 30 MHz. Lebar pita spectrum GSM900 sebesar 15 MHz dan penomoran kanal ARFCN-nya dimulai dari 0 dan seterusnya, dengan lebar pita per kanal GSM adalah 200 kHz (0,2 MHz). Maka, jumlah kanal untuk GSM900 adalah 15 MHz dibagi 0,2 MHz yaitu sebanyak 75 kanal.

2.6 GPRS (*General Packet Radio Service*)

Struktur jaringan GPRS (*General Packet Radio Service*) adalah penambahan dua *node* jaringan baru dalam GSM yaitu SGSN (*Serving GPRS Support Node*) dan GGSN (*Gateway GPRS Support Node*). Secara umum GPRS adalah suatu teknologi *packet switch* yang memungkinkan pengiriman dan penerimaan data lebih cepat jika dibandingkan dengan penggunaan teknologi *circuit switch data* atau CSD [7].

Komponen-komponen utama jaringan GPRS adalah sebagai berikut [7]:

1. SGSN, yaitu gerbang penghubung jaringan BSS/BTS ke jaringan GPRS.
2. GGSN, yaitu gerbang penghubung jaringan GSM ke jaringan internet.
3. PCU, yaitu komponen di level BSS yang menghubungkan terminal.

Packet switching bekerja dengan mentransmisikan data yang dibagi menjadi bagian-bagian kecil (paket) lalu diubah kembali menjadi data semula. *Packet switching* dapat mentransmisikan ribuan hingga jutaan paket per detik. Selain itu, memungkinkan untuk pemakaian kanal transmisi secara bersamaan oleh pengguna lain [7].

2.7 Enkripsi Pada GSM

Selama terjadi komunikasi pada jaringan, semua data pengguna (seperti pesan teks dan panggilan) yang dipertukarkan melalui media udara dienkripsi terlebih dahulu untuk menjaga kerahasiaan data [12].

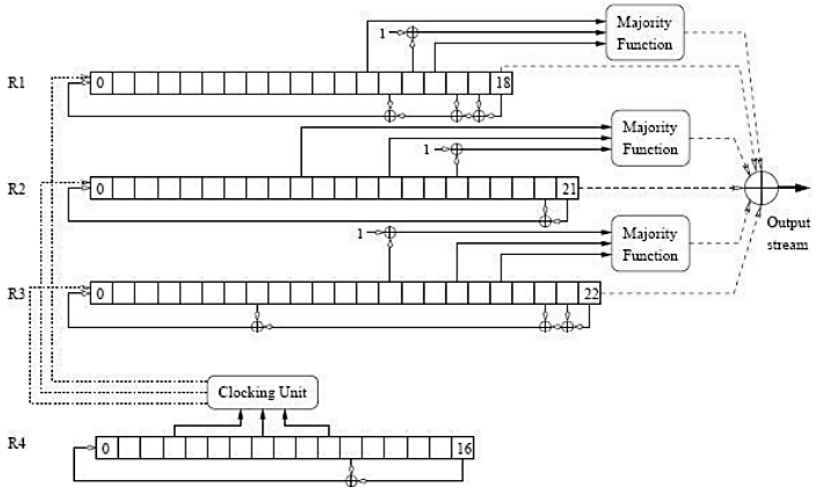
Sistem GSM menggunakan kriptografi simetri, karena menggunakan sebuah kunci privat yaitu KC (*Key Ciphering*). Kunci Kc digunakan untuk enkripsi dan juga dekripsi. Kunci Kc hanya dapat diketahui oleh perangkat (ponsel) dan jaringan [12].

Algoritma yang digunakan untuk proses enkripsi pada saat komunikasi data berlangsung adalah algoritma A5. Berbeda dengan algoritma A3, deskripsi algoritma A5 merupakan bagian dari GSM walaupun algoritma A5 tidak dibuat publik [12].

Terdapat dua versi algoritma A5 yang sering digunakan dalam GSM, yaitu A5/1 dan A5/2 yang merupakan *stream cheaper*. Selain itu, terdapat tambahan versi baru yang telah distandardisasikan tetapi belum digunakan di jaringan GSM yaitu algoritma A5/3. Algoritma A5/3 didasarkan pada *block-cipher* pada algoritma KASUMI. Algoritma A5/1 dan A5/2 merupakan algoritma pertama yang ditentukan standarnya oleh GSM dan dirancang berdasarkan sistem kontrol *clocking* LFSRs yang sederhana [12].

Algoritma A5/2 terdiri dari empat LFSR (*Linear Feedback Shift Register*) dengan panjang maksimum yaitu: R1, R2, R3, dan R4. Register-register tersebut

mempunyai panjang 19 bit, 22 bit, 23 bit, dan 17 bit. Contoh: pada saat R4 dikunci berdasarkan mekanisme penguncian (*clocking*), nilai XOR R4[17-0-1=16] dan R4[17-5-1=11] dihitung, kemudian registernya digeser satu bit ke kanan dan nilai hasil XOR tersebut ditempatkan di R4[0] [12].



Gambar 2.7 Algoritma A5/2 [12]

Pada algoritma A5/2, R1, R2, dan R3 dikunci dilakukan berdasarkan mekanisme penguncian dengan aturan seperti pada Gambar 2.6 yaitu R4 mengontrol penguncian R1, R2, dan R3. Saat penguncian terhadap R1, R2, dan R3 dilakukan, bit-bit R4[3], R4[7], dan R4[10] merupakan *input* dari unit penguncian. Unit pengujian tersebut melakukan sebuah fungsi mayoritas pada bit-bit yang ada. R1 dikunci jika dan hanya jika R4[10] sesuai dengan mayoritas. R2 dikunci jika dan hanya jika R4[3] sesuai dengan mayoritas. R3 dikunci jika dan hanya jika R4[7] sesuai dengan mayoritas. Setelah penguncian-penguncian terhadap register R1, R2, dan R3 dilakukan, kemudian R4 dikunci [12].

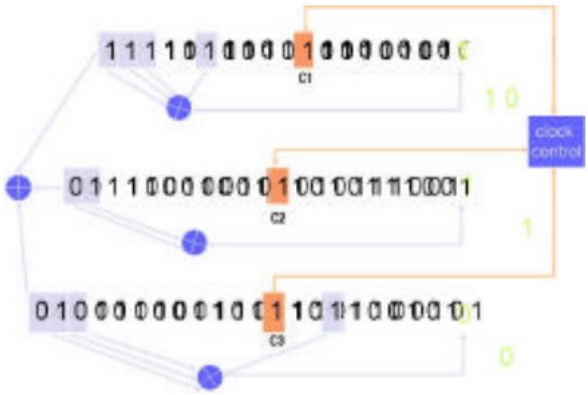
Setelah proses penguncian dilakukan, satu bit *output* telah siap untuk dihasilkan pada A5/2. Bit *output* merupakan fungsi non linier dari status internal R1, R2, dan R3. Setelah dilakukan inisialisasi 99 bit *output* dibuang dan 228 bit berikutnya digunakan sebagai *output key-stream* [12].

Sedangkan proses pembangkitan *key-stream* adalah [12]:

1. Inisialisasi status internal dengan nilai Kc dan jumlah *frame*.
2. Isi nilai bit-bit 1[15], R2[16], R3[8], dan R4[10] dengan 1.
3. Jalankan algoritma A5/2 untuk 99 *clocks* dan abaikan *output*-nya.

4. Jalankan algoritma A5/2 untuk 228 *clocks* berikutnya dan gunakan *output*-nya sebagai *key-stream*.

Pada dasarnya, algoritma A5/2 dibangun dengan kerangka yang sama dengan A5/1. Fungsi-fungsi *feedback* untuk register R1, R2, dan R3 pada A5/2 sama dengan fungsi *feedback* pada A5/1, begitu juga dengan proses inisialisasi yang dilakukan A5/1 dan A5/2. Yang membedakan algoritma A5/1 dan A5/2 adalah A5/1 hanya terdiri dari tiga LFSR dengan panjang maksimum masing-masing R1, R2, R3 adalah 19 bit, 22 bit, dan 23 bit, sehingga tidak ada pendefinisian untuk register R4, maka A5/2 juga harus melakukan inisialisasi R4 dan nilai satu bit pada tiap register harus diisikan dengan nilai 1 setelah dilakukan inisialisasi. Selain itu, A5/2 membuang 99 bit *output* sementara A5/1 membuang 100 bit *output* [12].

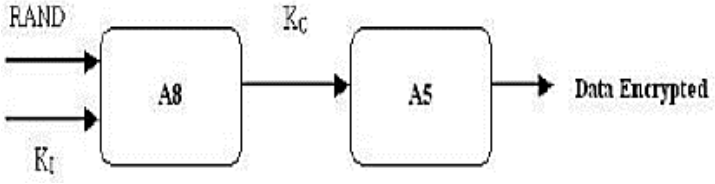


Gambar 2.8 Struktur Algoritma A5/1 [12]

Mekanisme enkripsi data adalah sebagai berikut [12]:

1. Memproses RAND, yang diterima pada saat akan melakukan otentikasi pengguna, dengan algoritma A8 dan Ki untuk menghasilkan kunci enkripsi Kc (*ciphering key*).
2. Mengenkripsi *plaintext* dengan algoritma A5 dan kunci Kc untuk menghasilkan *ciphertext*, yang akan ditransmisikan melalui jaringan.

Skema untuk enkripsi ditunjukkan pada Gambar 2.8 berikut.



Gambar 2.9 Skema Enkripsi Dalam GSM [12]

Pada saat algoritma A3 dijalankan pada proses otentikasi pengguna, pada saat yang sama algoritma A8 juga dijalankan. Kc dibangkitkan pada saat dilakukan otentikasi pengguna. Untuk setiap panggilan, Kc yang dibangkitkan akan berbeda nilainya. Kc hasil proses algoritma A8 disimpan ke dalam SIM dan terbaca oleh ponsel. Jaringan juga membangkitkan Kc dan mendistribusikannya kepada *base station* (BTS) yang menangani koneksi [12].

2.8 Mail Server

Email merupakan protokol pengiriman pesan elektronis melalui jaringan komputer, baik pada jaringan lokal area maupun internet. Untuk dapat membuat dan mengirim email, yang diperlukan adalah suatu program mail *client.* Dalam proses pengiriman, email akan melalui beberapa poin sebelum akhirnya sampai di alamat email tujuan. Seperti ditunjukkan pada diagram berikut [13].

Email dibuat → Mail *client* (pada komputer atau alamat email *user* → SMTP server penyedia email *user* → MTA server penyedia email *user* → Internet → MTA server penyedia email penerima → SMTP dan POP3/IMAP server penyedia email penerima → Mail *client* (pada komputer/alamat email penerima) → Email dibaca oleh penerima [13].

Berdasarkan diagram tersebut terdapat tiga komponen penting agar email dapat dikirim dan diterima oleh *user*, di antaranya SMTP (*Simple Mail Transfer Protocol*), POP3 (*Post Office Protocol version 3*)/IMAP (*Internet Message Access Protocol*), dan MTA (*Mail Transfer Agent*) [13].

2.8.1 SMTP (*Simple Mail Transfer Protocol*)

SMTP adalah suatu protokol yang digunakan untuk mengirimkan email dari komputer pengirim ke mail server penyedia email milik pengirim. Ketika sebuah email dikirimkan, komputer akan mengarahkan email tersebut ke sebuah SMTP server untuk diteruskan ke mail server tujuan. *Port* yang digunakan oleh SMTP adalah port 25 [13].

2.8.2 POP3 (*Post Office Protocol version 3*)/IMAP (*Internet Message Access Protocol*)

POP3 adalah protokol yang digunakan untuk mengambil email dari mail server penyedia email milik penerima. POP3 pada sebuah mail server berfungsi untuk menampung sementara email dari pengirim yang belum diambil oleh penerima yang berhak. Setelah email tersebut diambil, maka secara otomatis email akan terhapus dari mail server. *Port* yang digunakan oleh POP3 adalah *port* 110 [13].

2.8.3 MTA (*Mail Transfer Agent*)

MTA adalah salah satu komponen yang bertugas untuk memungkinkan email dapat dikirim dari satu mail server ke mail server lainnya. MTA juga berfungsi untuk mengatur koneksi antar mail server [13].

2.9 Postfix

Postfix dianggap sebagai MTA yang jauh lebih aman daripada Sendmail dan lebih cepat daripada Qmail. Pengamanan yang ada pada mail server Postfix adalah sebagai berikut [13].

1. Hak-hak istimewa terbatas

   Hampir semua program *daemon* Postfix dapat dijalankan dengan hak istimewa terbatas yang tetap dalam lingkungan *chroot*. Hal ini sangat mendukung untuk program-program yang muncul dalam jaringan: server SMTP dan *client* SMTP. Walaupun tidak ada jaminan kerja sama dengan sistem, fitur ini tetap dapat menambah keamanan sistem meskipun sedikit.

2. Penyekatan

   Postfix menggunakan proses-proses yang terpisah untuk menyekat kejadian satu sama lain. Hal ini terlihat dari tidak adanya jalur langsung dari jaringan ke program penyebar lokal (*security sensitive local delivery program*), sehingga pengganggu harus melewati banyak program lain terlebih dahulu.

Terdapat beberapa bagian Postfix yang bersifat multiproses (*multi-threaded*). Akan tetapi, semua program yang berhubungan dengan dunia luar tetap bersifat tunggal (*single-threaded*). Proses-proses terpisah ini memberikan penyekatan yang lebih baik daripada multiproses dalam satu ruang yang dibagi-bagi.

3. Lingkungan yang terkontrol

   Dalam proses pengiriman email, tidak ada program yang berjalan di bawah kendali *user*. Namun sebagai penggantinya, program-program Postfix berjalan dalam sebuah *daemon master* yang tetap berjalan dalam sebuah lingkungan yang terkendali, tanpa adanya hubungan *parent-child* dengan proses *user*. Keuntungannya, pendekatan ini dapat menurunkan penggunaan atribut-atribut pada proses UNIX/Linux yang lain, sinyal, serta pembukaan *file-file* dan variabel lingkungan sistem UNIX, sehingga mengurangi kemungkinan *parent* yang berbahaya melewati *child*.

4. Set-uid

   Semua program Postfix diatur dengan set-uid. Konsep uid ini muncul karena adanya kesalahan pada sistem UNIX, dimana set-uid dan set-gid awalnya diharapkan dapat memberikan pengaruh yang bagus pada sistem UNIX, tetapi ternyata yang didapatkan sebaliknya. Setiap kali ada program baru yang dimasukkan ke sistem UNIX, set-uid selalu menimbulkan masalah keamanan seperti program *shared libraries*, sistem *file*/proc, dan dukungan beragam bahasa. Hal ini dikarenakan set-uid tidak dapat mengenali fitur-fitur tersebut.

   Pada awalnya, direktori *maildrop* mempunyai atribut *write*, sehingga proses-proses lokal dapat mengirimkan email tanpa menghiraukan set-uid atau set-gid atau dari proses *mail daemon*. Padahal, seharusnya direktori *maildrop* tidak digunakan untuk email yang datang dari jaringan dan *file-file* antriannya tidak beratribut *write* bagi *user* yang tidak mempunyai hak. Direktori yang memiliki atribut *write* ini ternyata memberikan celah keamanan dimana *user* lokal dapat mengakses *file-file maildrop* orang lain

sehingga *file-file* itu dapat diakses atau disebarkan. *User* lokal juga dapat mengisi direktori *maildrop* dengan sampah (email yang dibuang atau tidak berguna) dan mungkin juga dapat berusaha membuat agar sistem mengalami kerusakan. Selain itu mereka juga dapat memindahkan *file-file* orang lain ke dalam direktori *maildrop* dan menyebarkannya sebagai email. Walaupun demikian *file-file* antrian Postfix memiliki format tertentu dimana salah satu di antara 10^{12} *file-file* non-Postfix akan dikenali sebagai *file* antrian Postfix yang sah.

Karena kemungkinan-kemungkinan timbulnya bahaya akibat direktori *maildrop* yang memiliki atribut *write* ini maka Postfix menggunakan program bantuan set-gid postdrop untuk perizinan mail.

5. Kepercayaan

   Program-program Postfix tidak akan mudah memercayai isi dari *file-file* antrian atau pesan IPC internal Postfix. *File-file* antrian ini tidak mempunyai catatan pada *disk* untuk disebarkan pada tujuan-tujuan tertentu seperti ke *file* atau perintah program. Sebagai gantinya, program-program seperti agen pengiriman lokal yang akan berusaha menghasilkan keputusan strategis berdasarkan informasi tangan pertama.

   Program-program Postfix juga tidak akan memercayai data yang berasal dari jaringan, karena secara teknis Postfix akan melakukan penyaringan data yang dikirim berikut pengirimnya terlebih dahulu sebelum menyebarkannya melalui variabel internet. Hal yang juga merupakan suatu pelajaran adalah hal keamanan, yaitu jangan biarkan data apapun yang berasal dari jaringan mendekati *shell* tanpa ada penyaringan data terlebih dahulu.

6. *Input* yang besar

   Pengalokasian *string* dan *buffer* pada memori dilakukan secara dinamis, hal ini berguna untuk mencegah terjadinya permasalahan *overrun* pad *buffer.*

   Setiap *input* pesan yang mempunyai baris yang panjang akan dipecah atau dipotong-potong menjadi beberapa urutan yang saling berhubungan dengan ukuran yang lebih kecil. Pecahan ini kemudian akan disatukan kembali pada

saat pengiriman. Untuk mencegah terjadinya *overrun buffer* pada sistem operasi yang lebih lama, maka diagnosis pemotongan ini dilakukan dalam satu tempat kemudian dikirim ke *interface syslog.* Meskipun demikian, tidak ada cara umum yang digunakan untuk memotong data sebelum dilewatkan ke *system call* atau *library routine.* Pada beberapa sistem operasi, *software* mungkin masih mengalami masalah *overrun buffer* karena kerentanan *software* utamanya.

7. Pengamanan-pengamanan yang lain

   Agar sistem mail tidak menjadi padat karena beratnya beban, jumlah tipe-tipe objek yang menggunakan memori pun dibatasi. Selain itu untuk mencegah hal-hal yang lebih buruk lagi, maka setiap terjadi masalah, *software* akan menghentikan pengiriman untuk beberapa lama sebelum terjadinya *error* yang fatal atau sebelum me-*restart* program yang gagal.

2.10 RTL-SDR

RTL-SDR adalah USB *dongle* yang digunakan sebagai pemindai radio berbasis komputer untuk menerima sinyal radio secara langsung. *Frequency range* RTL-SDR Rafael Micro R820T/2 adalah sebesar 24-1766 MHz (dapat dikembangkan menjadi 13 - 1864 MHz dengan teknik tertentu) [14].

Gambar 2.10 RTL2832U [15]

Maksimum *sample rate* sebesar 3.2 MS/s (mega sampel per detik). Akan tetapi, RTL-SDR tidak stabil pada tingkat tersebut dan memungkinkan dapat terjadinya *drop samples*. *Sample rate* saat tidak terjadi *drop samples* sebesar 2.56

MS/s, namun pada 2.8 MS/s dan 3.2 MS/s dapat bekerja dengan baik menggunakan USB 3.0 [14].

Drop samples masih dapat bekerja dengan baik jika digunakan untuk visualisasi spektrum, tetapi akan bermasalah jika digunakan untuk demodulasi atau me-*decode* sinyal [14].

2.11 Penggunaan TMSI (*Temporary Mobile Subscriber Identity*)

Temporary Mobile Subscriber Identity (TMSI) ditentukan nilainya pada saat *International Mobile Subscriber Identity* (IMSI) ditransmisikan ke AuC, yaitu saat *Subscriber Identity Module* (SIM) diaktifkan untuk pertama kalinya. TMSI digunakan oleh MS untuk melapor kepada jaringan atau pada saat inisialisasi panggilan. Pada saat MS dimatikan, TMSI disimpan pada SIM untuk dipakai kembali pada waktu mendatang. Sedangkan jaringan menggunakan TMSI untuk berkomunikasi dengan MS. TMSI dikirim kepada MS setelah prosedur otentikasi pengguna dilakukan. Pemetaan TMSI ke IMSI yang berkoresponden, dilakukan oleh jaringan, tepatnya ditangani oleh VLR. TMSI hanya valid di suatu *Location Area* (LA) tertentu. TMSI di-*update* setidaknya setiap perubahan lokasi (ketika ponsel berganti LA atau setelah periode tertentu). TMSI juga dapat diubah kapanpun oleh jaringan. TMSI dikirimkan dalam bentuk cipher [12].

2.12 Kali Linux

Kali linux merupakan sebuah sistem operasi *open source* yang dikelola dan dibiayai oleh Offensive Security, pengembang kelas dunia pada pelatihan keamanan informasi dan layanan *penetration testing*. Offensive Security juga mengelola Exploit Database dan kursus Metasploit Unleashed untuk Kali Linux [16].

2.13 Linux Ubuntu

Sistem operasi Ubuntu adalah suatu variasi pada sistem operasi *desktop* Ubuntu Linux, yang kode sumber terbuka (*open source*) dan telah menjadi sistem operasi standar di seluruh dunia. Lembaga pemerintahan keamanan nasional Inggris

(CESG) menilai Ubuntu sebagai sistem operasi yang paling aman dari 11 sistem operasi yang diuji [17].

2.14 Gr-GSM

Gr-GSM adalah *tool* yang berfungsi sebagai *gsm-receiver* dimana diprogram oleh Piotr Krysik (juga sebagai penulis *gr-gsm*) untuk proyek Airprobe. Tujuannya adalah untuk mengolah seperangkat alat untuk menerima informasi yang dikirimkan oleh perangkat GSM [18].

2.15 Tinjauan Mutakhir

Penelitian ini mengacu terhadap beberapa referensi yang terkait dengan penelitian sebelumnya, dimana masing-masing peneliti menggunakan metode yang berbeda sesuai dengan permasalahan yang dibahas. Berikut ini beberapa referensi yang digunakan untuk membedakan pembahasan yang dibahas pada penelitian yang telah ada.

Tabel 2.1 Tinjauan Mutakhir

Penulis	Judul	*Software*	*Hardware*	Hasil Penelitian
[3]	Analisis Tingkat Keamanan Jaringan Sinyal GSM Menggunakan Metode *Tapping* Pada Jaringan GSM	Wireshark	RTL-SDR	Informasi *provider*, GSM *Frame Number*, IMSI (*International Mobile Subscriber Identity*), TMSI (*Temporary IMSI*), dan *burst frame* (data komunikasi) GSM.
[4]	Implementasi Gr-GSM Untuk *Decoding* Komunikasi GSM Terenkripsi	Linux Ubuntu 14.04 LTS, *Driver* RTL-SDR 2832U, Wireshark, Gr-GSM, Kalibrate, GNURadio, Airprobe, GSM Framecoder, Kalibrate	RTL-SDR, *SIM Card Reader*	Penangkapan sinyal dan *decoding* mendapatkan informasi operator dari frekuensi berupa *GSM Frame Number*, IMSI (*International Mobile Subscriber Identity*), TMSI (*Temporary* IMSI), algoritma keamanan yang digunakan pada operator, dan data komunikasi pada keamanan GSM.

[19]	Implementasi GNURadio GR-DVBT Untuk *Decoding* Sinyal Televisi Digital	GNURadio, Linux Ubuntu 14.04 LTS, *Driver* RTL-SDR R2832U, GR-DVBT	RTL-SDR, USRP N210	Semua bentuk komunikasi video dapat dikirim melalui sinyal pada frekuensi tertentu.
[20]	*Sniffing GSM Traffic Using RTL-SDR And Kali Linux OS*	Kali Linux, Wireshark, Airprobe, Kalibrate, GNURadio	RTL-SDR *Dongle*	Informasi mengenai identitas pengguna seluler dapat ditangkap dari sinyal GSM, informasi yang diperoleh adalah informasi pada SIM seperti IMSI dan lokasi pengguna.
[21]	Analisis Sinyal Komunikasi UAV Menggunakan SDR	Ubuntu 16.04 LTS, *Universal Radio Hacker* (URH), GQRX, HackRF	HackRF One, UAV Syma X5HW, *remote control*	Informasi pergerakan UAV berupa *throttle*, *yaw*, *pitch* dan *roll* dapat diketahui dengan menggunakan perangkat SDR; Panjang 1 *frame* sinyal informasi yang didapatkan yaitu sepanjang 144 bit atau 18 byte, yang di dalamnya terdapat *preamble*, *address*, *payload* dan CRC.

[22]	*Sniffing* Sinyal GSM Menggunakan RTL-SDR Untuk Menentukan Koordinat Pengguna GSM	GNURadio, Wireshark, Phone Tracker	RTL-SDR	Data yang dihasilkan dalam proses *sniffing* sinyal GSM yang berupa *Local Area Identification* (LAI) dan *Cell identification* (*Cell id*). Data-data tersebut yang dapat menunjukan koordinat pengguna sinyal GSM yang ter-*sniffing* dengan bantuan program aplikasi phone tracker.
[23]	Implementasi GNURadio GR-DVBT2 Untuk *Decoding* Sinyal Televisi Digital	GNURadio, Linux Ubuntu 16.04	RTL-SDR	Receiver RTL-SDR dapat mendeteksi sinyal televisi digital dan mampu menghasilkan *output* berupa suara.
[24]	*Sniffing* Sinyal GSM dengan RTL-SDR, GNU Radio dan Wireshark	GNURadio, Wireshark, Phone Tracker	RTL-SDR	Data yang dihasilkan dalam proses *sniffing* sinyal GSM yaitu berupa *Local Area Identification* (LAI). Percobaan *sniffing* sinyal GSM yang dilakukan berberapa kali hanya mendapatkan satu pengguna GSM

				yang ter-*sniffing* dan dengan koordinat lokasi yang sama.
[25]	*GSM Wireless Sniffer using Software Defined Radio*	Kali Linux, GQRX, Wireshark, Airprobe	RTL-SDR	Dengan decoding BCCH & CCCH. Metode ini mengeksploitasi kelemahan GSM yang mengambil redundansi dari paket BCCH & CCCH yang menjalani pengodean paket. Selama redundansi menghasilkan pola yang diketahui, potongan berulang dapat dieksploitasi bersama dengan data *chipper*. Setelah operasi "xor" antara keduanya, *output* dapat mengarah ke Ki-*sequence* yang digunakan untuk mengenkripsi seluruh paket.