

BAB II

LANDASAN TEORI

2.1 GPS

GPS adalah suatu sistem navigasi global yang melingkupi seluruh planet bumi ini. Sistem ini dibentuk oleh 24 satelit GPS dan stasiun-stasiun di bumi. Fungsi dari satelit-satelit tersebut adalah untuk memberikan informasi pada seseorang yang berada di bumi, seperti jarak antara orang tersebut dengan satelit, waktu internasional dan lain sebagainya. Sedangkan stasiun-stasiun di bumi digunakan untuk memantau data-data yang akan dipancarkan oleh satelit sekaligus mengoreksi adanya penyimpangan-penyimpangan yang terjadi.



Gambar 2.1. Satelit GPS

Secara sederhana cara kerja GPS dapat dijelaskan sebagai berikut. GPS penerima di bumi menangkap sinyal dari tiga satelit GPS. Dengan menggunakan rumus fisika sederhana, (Global Positioning System Overview, 2003)

$$\text{Jarak} = \text{Kecepatan Cahaya} \times \text{Waktu} \quad (2.1)$$

maka jarak masing-masing satelit ke penerima dapat diketahui. Dengan diketahui tiga jarak tersebut maka bila masing-masing jarak itu adalah jari-jari lingkaran

akan didapat tiga buah lingkaran yang berpotongan di satu titik. Titik potong itulah yang merupakan posisi seseorang yang sebenarnya di atas bumi.

Keterangan di atas merupakan cara kerja sederhana dari GPS apabila dianggap bahwa bumi itu datar. Untuk menghasilkan posisi yang baik secara 3 dimensi maka diperlukan minimal 4 buah satelit. Semakin banyak satelit yang dapat diterima oleh GPS *receiver* maka semakin akurat posisi yang didapatkan.

Selain peran dari satelit-satelit, GPS juga harus terus menerus dipantau oleh stasiun – stasiun di bumi. Hal ini perlu sekali dilakukan karena satelit-satelit tersebut letaknya jauh dari permukaan bumi sehingga kesalahan akan mungkin sekali terjadi.

Contoh dari kesalahan itu adalah bergesernya posisi satelit dari asalnya. Apabila hal ini terjadi, walaupun pergeserannya sangat kecil tetapi hal itu akan mengakibatkan perbedaan yang sangat besar ketika dibaca di bumi mengingat jarak antara penerima dengan satelit begitu jauh. Selain itu stasiun bumi juga akan mengkoreksi referensi waktu yang dipakai dalam sistem GPS ini sehingga didapatkan suatu *timing* yang tepat. Jadi, dengan adanya stasiun-stasiun bumi ini maka faktor kesalahan masing-masing satelit dapat dipantau dan dikirimkan ke tiap-tiap penerima di bumi sehingga masing-masing penerima dapat mengkoreksi kesalahan yang terjadi.

Pusat pengendali GPS oleh Amerika Serikat ditempatkan di pangkalan udara Falcon di Colorado. Selain itu juga di tempatkan stasiun-stasiun bumi yang lain, misalnya di Hawaii dan di tempat-tempat lain seperti pada gambar 2.2. Setiap stasiun pemantau akan selalu mengecek tiap-tiap satelit GPS di wilayahnya. Apabila suatu saat ternyata terjadi penyimpangan pada satelit maka stasiun itu

akan segera melakukan pengkoreksian sistem ke satelit dengan jalan mengirimkan faktor *error* yang terjadi saat itu.



Gambar 2.2. Stasiun Bumi Pengendali Satelit-Satelit GPS

2.1.1 Cara kerja GPS

Gambaran singkat tentang sistem ini dalam memberikan informasi-informasi posisi kepada pembawa GPS *receiver* yang berada di muka bumi. Dalam bahasan selanjutnya akan diterangkan secara lebih detail cara kerja dari sistem GPS itu sendiri.

Pada dasarnya cara kerja sistem GPS mempunyai 5 titik dasar (Global Positioning System Overview, 2003), yaitu sebagai berikut :

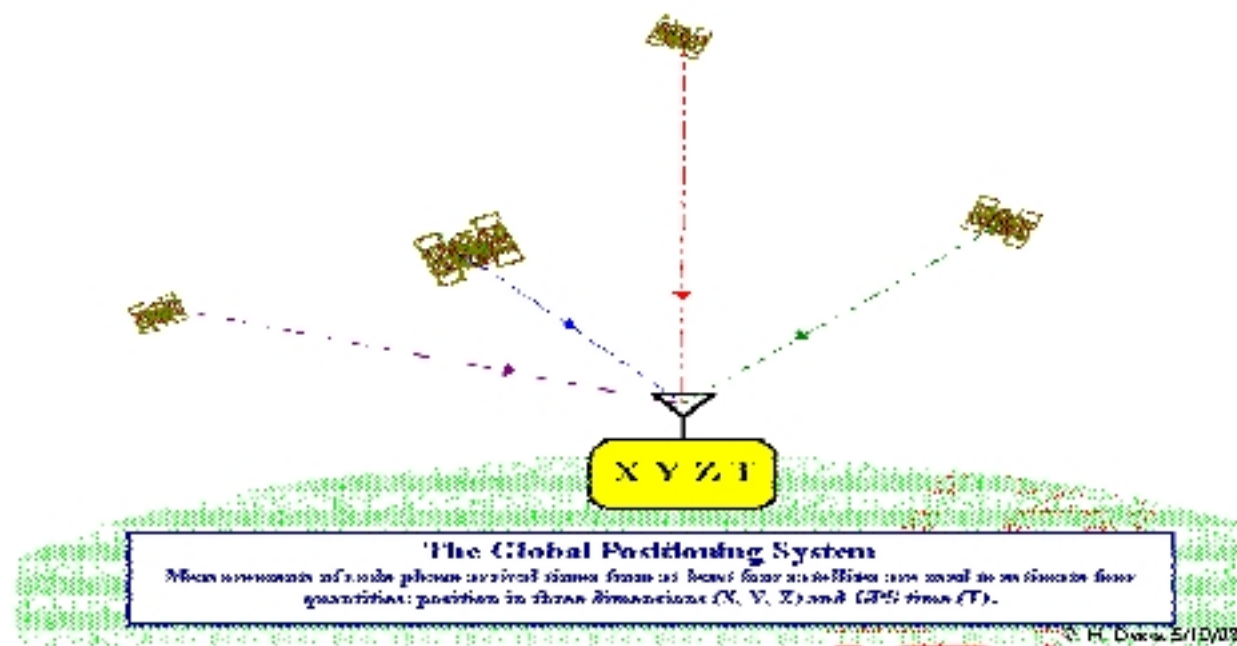
1. Dasar dari GPS adalah "Pertemuan Titik Potong" dari satelit-satelit GPS.
2. Untuk menemukan titik potong dari satelit-satelit GPS tersebut, pesawat penerima harus mengukur jarak masing-masing satelit ke *receiver* dengan cara menghitung jarak tempuh sinyal radio dari satelit ke *receiver*.

3. Untuk menghitung jarak tempuh sinyal radio ke *receiver* maka GPS memerlukan sistem *timing* yang sangat akurat sehingga waktu pengiriman dengan waktu penerimaan dapat diketahui dengan tepat.
4. Selain itu untuk menghitung jarak maka harus mengetahui dengan pasti posisi masing-masing satelit sehingga dengan sedikit perhitungan matematika dapat ditentukan posisi titik potong itu yang tidak lain adalah posisi penerima.
5. Yang terakhir adalah pengkoreksian hasil yang diterima dengan faktor kesalahan yang terjadi selama gelombang radio dikirimkan, sebagai contoh *delay* yang terjadi akibat gelombang radio melewati *troposfer* dan faktor pembelokan saat gelombang radio melalui lapisan *ionosfer*.

Selanjutnya akan dijelaskan pengertian dari masing-masing titik yang telah dijelaskan di atas sehingga dapat membentuk rangkaian cara kerja GPS secara lengkap.

2.1.2 Mencari titik potong satelit

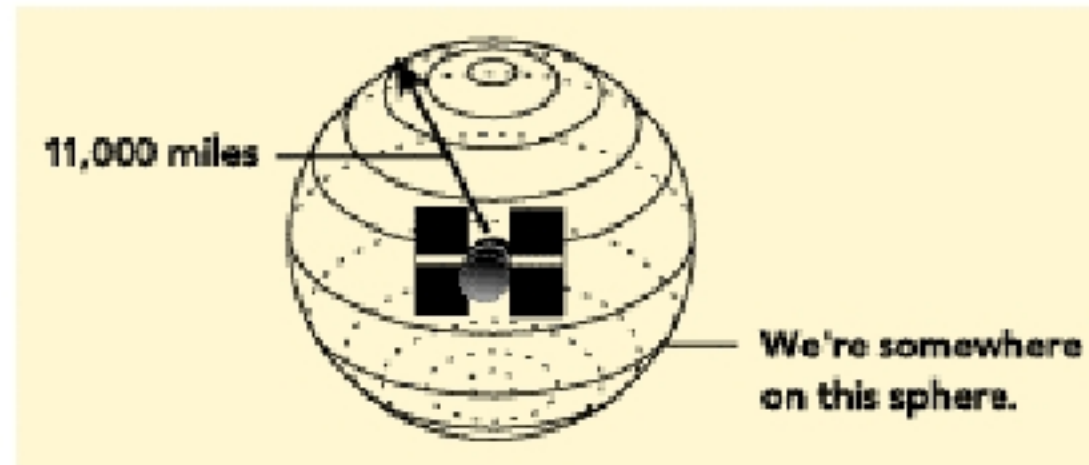
Apabila pertama kali dilihat bagaimana GPS dapat memberitahu posisi seseorang secara tepat di atas bumi, maka dirasakan bahwa sistem yang dipakai sangatlah rumit dan sulit untuk dimengerti. Akibatnya seseorang akan mengabaikan cara kerja dari GPS itu sehingga akhirnya orang itu hanya menjadi *user* saja tanpa tahu sedikitpun mengenai cara kerja dan ide dasar dari GPS tersebut.



Gambar 2.3 Mencari Titik Potong di Bumi

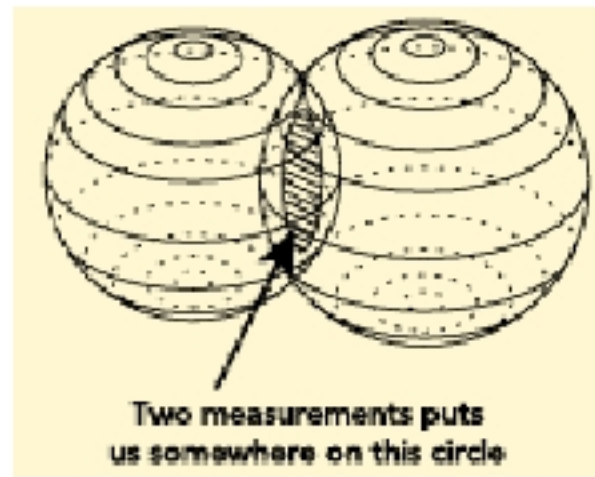
Sebenarnya, dibalik semuanya itu, GPS hanya didasarkan pada satu ide saja, yaitu “menjadikan satelit sebagai titik referensi untuk mengetahui posisi seseorang yang ada di atas bumi”. Dengan demikian dapat diketahui jarak dan posisi masing-masing satelit terhadap pembawa GPS *receiver*. Setelah itu dengan mengambil data dari beberapa satelit dapat diperoleh titik potong dari jarak satelit-satelit tersebut sehingga diperoleh koordinat yang tepat seperti pada gambar 2.3.

Sebagai contoh, apabila suatu pesawat penerima dapat menerima sinyal dari 3 buah satelit GPS. Kemudian dengan berbagai perhitungan akhirnya didapatkan bahwa jarak penerima dengan satelit pertama adalah 11.000 mil. Jarak 11.000 mil akan mempunyai banyak sekali kemungkinan. Bila dianggap satelit pertama sebagai pusat dari bola dengan jari-jari 11.000 mil maka posisi penerima adalah sebuah titik di permukaan bola tersebut. Pada saat ini masih belum bisa didapatkan posisi yang sebenarnya karena faktor kemungkinannya masih terlalu luas.



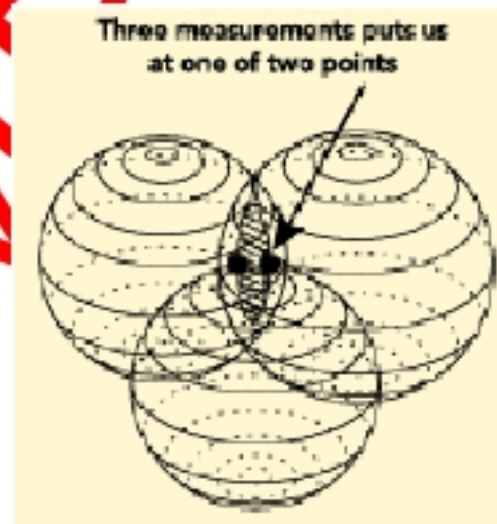
Gambar 2.4. Satu Satelit Menghasilkan Kemungkinan Posisi Pada Permukaan Bola

Selanjutnya, dari satelit kedua didapatkan bahwa jarak penerima dengan satelit kedua adalah 12.000 mil. Maka seperti pada satelit pertama pada gambar 2.4, akan didapatkan sebuah bola lagi dengan jari-jari 12.000 mil. Dan posisi penerima juga merupakan suatu titik di permukaan bola dengan jari-jari 12.000 mil tersebut. Dengan demikian telah didapatkan dua buah bola yang saling berpotongan seperti pada gambar 2.5. Perpotongan kedua bola tersebut akan menghasilkan sebuah lingkaran. Dengan demikian maka penerima adalah suatu titik di dalam lingkaran tersebut. Pada saat ini pun masih belum bisa didapatkan kepastian tentang letak posisi penerima yang sebenarnya karena faktor kemungkinannya masih terlalu luas. Namun demikian faktor kemungkinan tersebut sudah lebih sempit dari faktor kemungkinan yang pertama.



Gambar 2.5 Dua Satelit Menghasilkan Kemungkinan Posisi Pada Tepian Lingkaran

Dan dari satelit ketiga misalnya didapatkan jarak 13.000 mil sehingga akan didapatkan sebuah bola lagi. Dengan demikian telah didapatkan 3 buah bola yang saling berpotongan seperti pada gambar 2.6. Dari gambar di bawah dapat dilihat bahwa perpotongan ketiga bola tersebut akan menghasilkan dua buah titik dan posisi penerima adalah di salah satu titik potong tersebut. Dengan demikian posisi penerima semakin jelas.



Gambar 2.6 Tiga Satelit Menghasilkan Kemungkinan Posisi Pada Dua Titik

Apabila dilihat uraian di atas maka seharusnya bila diterima satu buah satelit lagi maka akan ditemukan titik posisi penerima yang sebenarnya. Namun hal itu tidak sederhana itu karena adanya beberapa faktor-faktor yang mempengaruhi, misalnya faktor *error* yang terjadi pada sistem GPS baik yang

terjadi karena alam maupun yang memang dengan sengaja dibentuk oleh Amerika untuk kepentingan militer.

2.1.3 Pengukuran jarak penerima dengan satelit

Dari titik uraian diatas telah dimengerti bagaimana GPS dapat menentukan posisi seseorang secara tepat di bumi. Namun yang masih menjadi pertanyaan adalah bagaimana caranya mendapatkan jarak yang tepat antara penerima dengan satelit-satelit tersebut. Hal itu dapat dijelaskan sebagai berikut.

Sebagai contoh, apabila sebuah mobil berangkat dari kota X pada jam 6 pagi dan tiba di kota Y pada jam 7 pagi itu juga. Mobil berjalan dengan kecepatan rata-rata 100 km/jam. Maka waktu yang diperlukan adalah 1 jam. Jarak kota X dengan kota Y adalah $100 \text{ km/jam} \times 1 \text{ jam} = 100 \text{ km/Jam}$.



Gambar 2.7 Mencari Jarak Antara Satelit Dengan Penerima

Prinsip kerja dari GPS juga seperti itu. Dalam hal ini kecepatan yang dipakai adalah kecepatan *microwave* di udara yang di anggap 3.10^8 m/s yang merupakan konstanta kecepatan cahaya. Yang menjadi masalah di sini adalah mengenai waktunya. Untuk mengatasi masalah waktu ini, mula-mula satelit GPS mengirimkan sinyal informasi, dimana di dalam sinyal informasi tersebut terdapat

data yang diperlukan untuk sistem GPS. Yang paling penting adalah informasi tentang kapan sinyal tersebut dikirimkan. Sedangkan di pesawat penerima akan men-*decode* sinyal informasi itu dan mengambil informasi tentang waktu pengiriman sinyal seperti pada gambar 2.7. Kemudian informasi waktu tersebut dikurangkan kepada waktu sekarang yang ada di dalam penerima GPS. Akibatnya didapatlah selang waktu pengiriman dari satelit ke penerima GPS.

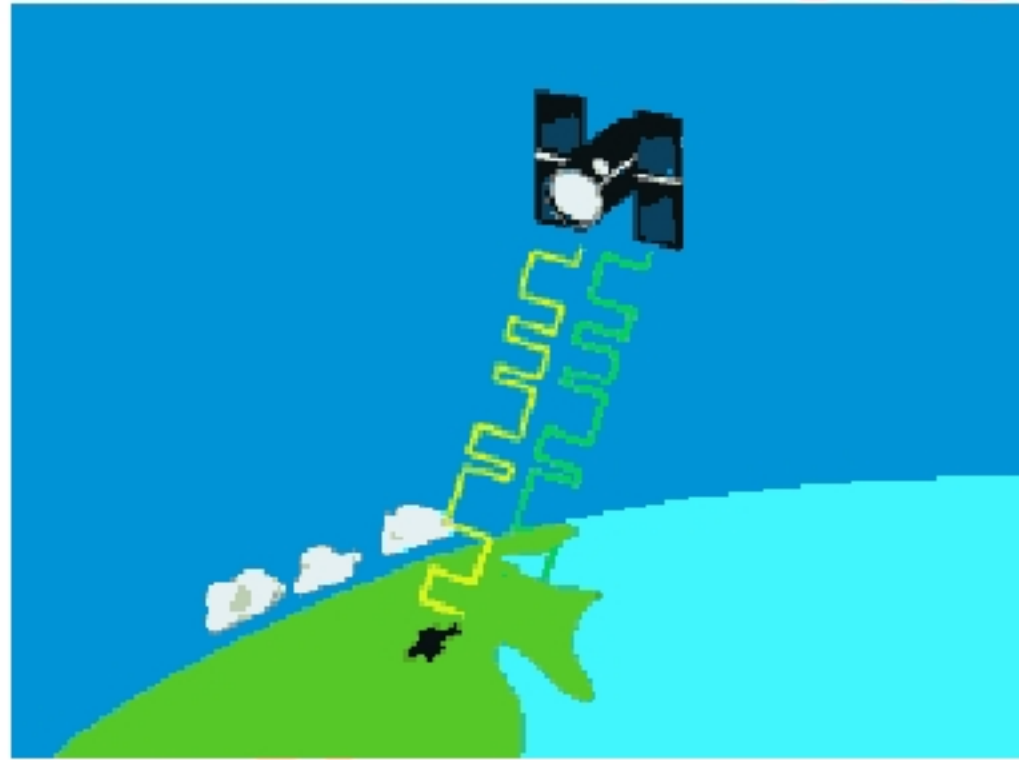
Untuk dapat menghasilkan perhitungan yang tepat maka antara satelit GPS dan penerima GPS harus mempunyai sistem pewaktu yang tepat sehingga data yang dikirimkan dapat tepat sama ketika diterima. Agar mempunyai ketepatan waktu yang sangat tinggi maka pada masing-masing satelit GPS menggunakan jam atom. GPS *receiver* yang ada di bumi akan menyamakan waktu yang dimiliki dengan waktu pada satelit. Dengan demikian maka ketepatan waktu GPS *receiver* dengan satelitnya menjadi tepat sehingga penghitungan selang waktu pengiriman dan penerimaan dapat diketahui dengan tepat pula.

2.1.4 Mendapatkan waktu yang tepat

Dalam pembahasan sebelumnya telah dikatakan bahwa untuk dapat mengakses informasi yang dipancarkan oleh satelit GPS harus disamakan terlebih dahulu waktu penerimaan dengan waktu tiap-tiap satelit. Setelah waktu GPS *receiver* sama dengan satelit GPS maka untuk selanjutnya yang bekerja adalah waktu yang ada di dalam GPS *receiver* itu sendiri untuk men-*decode* sinyal informasi yang lain.

Untuk dapat menghitung jarak antara *receiver* dengan satelit GPS maka diperlukan waktu gelombang radio yang dipancarkan oleh satelit yang

bersangkutan seperti pada persamaan 2.1. Untuk mendapatkan waktu yang sebenarnya maka harus dikurangi waktu yang diterima (waktu pengiriman) dengan waktu yang ada di dalam *receiver* saat itu. Dengan demikian jelaslah bahwa GPS *receiver* memerlukan suatu sistem *timing* yang benar-benar tepat, karena perbedaan sedikit saja akan menimbulkan kesalahan sampai ratusan kilometer.

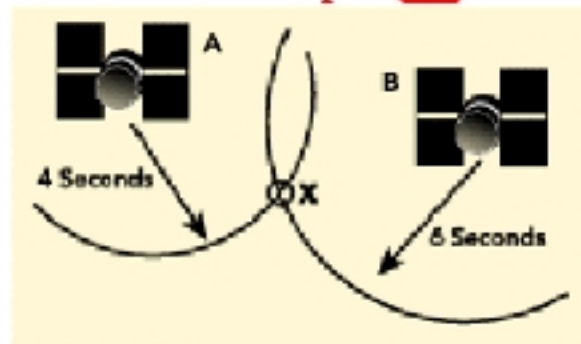


Gambar 2.8 Mendapatkan Waktu Yang Tepat

Masalah pewaktuan pada satelit GPS sudah tidak perlu lagi diragukan karena satelit GPS menggunakan jam atom sebagai sistem pewaktu-nya. Selain itu apabila sampai terjadi pergeseran (*offset*) maka akan di koreksi oleh stasiun yang ada di bumi. Untuk GPS *receiver* amat tidak mungkin menggunakan jam atom. Karena harga dari jam atom itu sendiri sangat mahal (USD 50.000 sampai USD 100.000). Oleh sebab itu sistem yang telah ada haruslah selalu mengadakan perhitungan terhadap *error* yang terjadi di setiap saat. Hasil yang didapatkan kemudian merupakan *offset* bagi GPS *receiver*. Selanjutnya *offset* tadi

dikurangkan kepada perhitungannya waktu yang sebelumnya., sehingga akan didapatkan sistem waktu yang setara dengan jam atom.

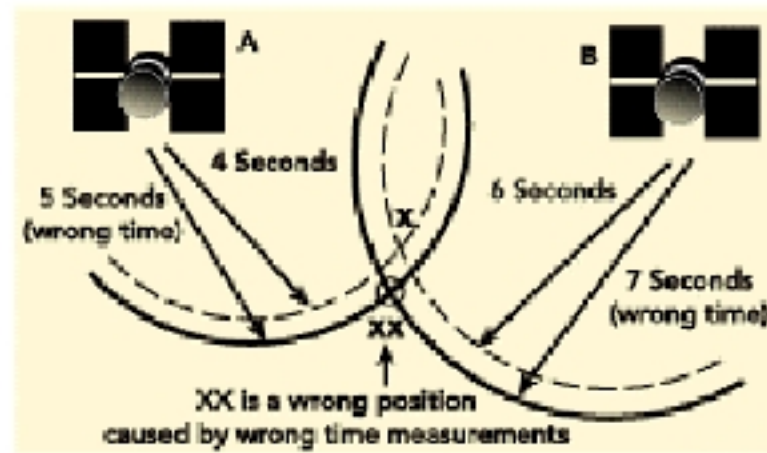
Sekarang akan dijelaskan terjadinya *error* seperti itu dan cara mengatasinya. Dianggap GPS bekerja secara 2 dimensi untuk mempermudah penjelasan. Suatu misal GPS *receiver* menerima sinyal informasi dari GPS A dan B. Dari satelit A sinyal diterima dalam waktu 4 detik dan dari satelit B sinyal diterima dalam waktu 6 detik. Kedua satelit akan membentuk dua buah lingkaran yang berpotongan di X yang mana X merupakan posisi penerima yang sebenarnya seperti pada gambar 2.9.



Gambar 2.9 Titik Potong Yang Terjadi Oleh 2 Buah Satelit Dalam Sistem 2 Dimensi

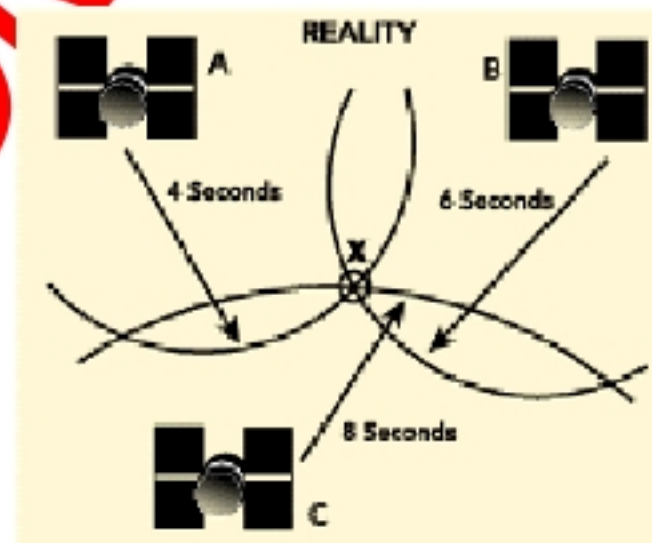
Bila terjadi terjadi pewaktu yang ada pada GPS *receiver* lebih lambat satu detik terhadap waktu GPS, akibatnya waktu yang dibutuhkan oleh sinyal dari satelit A menjadi 5 detik dan sinyal dari satelit B menjadi 7 detik. Hal itu tentu saja akan menyebabkan titik posisi yang dihitung akan bergeser. Misalkan saja titik bergeser ke XX.

Dari gambar 2.10 nampak *offset* yang terjadi adalah jarak dari X ke XX. Dengan menggunakan perhitungan matematika dapat dilakukan dengan cara mengurangi hasil yang dihitung pertama dengan *offset* yang didapatkan sehingga diperoleh sebuah titik X.



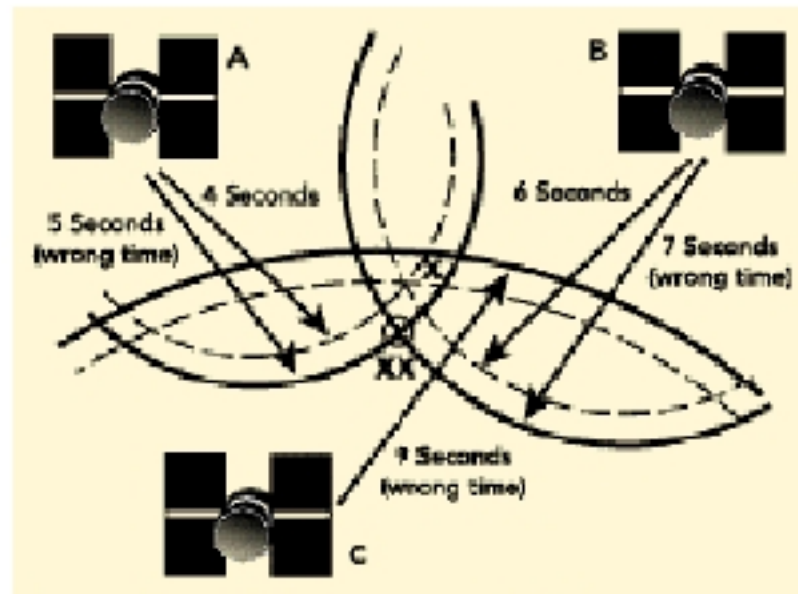
Gambar 2.10 Pergeseran Posisi Dari X ke XX Akibat Perhitungan Waktu Yang Salah

Kemudian diambil contoh lagi apabila perhitungan dilakukan dengan menggunakan tiga buah satelit, satelit A, satelit B dan satelit C. Dengan adanya ketiga satelit tersebut maka akan didapatkan tiga buah lingkaran yang berpotongan di satu titik. Seperti pada gambar 2.11 misalnya sebut titik itu sebagai X. Titik X akan menjadi titik potong yang sebenarnya, artinya secara ideal (tanpa gangguan) maka X merupakan titik koordinat dimana penerima berada saat itu.



Gambar 2.11 Titik Potong Yang Terjadi Oleh 3 Buah Satelit Dalam Sistem 2 Dimensi

Bila terjadi kesalahan *offset* sebesar satu detik pada masing-masing satelit maka akan didapatkan gambar seperti dibawah ini :



Gambar 2.12 Pergeseran Posisi Dari X ke XX Akibat Perhitungan Waktu Yang Salah

Nampak bahwa terjadi *offset* sebesar X sampai XX yang selanjutnya dipakai untuk mengkoreksi hasil perhitungan yang sedang dilakukan. Untuk selanjutnya garis *error* tebal pada gambar di atas disebut sebagai "*pseudorange*" yang digunakan dalam dunia GPS untuk mengatakan pengukuran yang mengandung kesalahan.

Titik XX merupakan perpotongan antara lingkaran A dan B. Dengan demikian lingkaran C tidak dapat memotong titik itu. Keadaan yang seperti inilah yang dapat digunakan sebagai *error indicator* bagi komputer bahwa kesalahan telah terjadi dan untuk selanjutnya dikoreksi. Pengkoreksian dapat dilakukan dengan berbagai cara menurut teori-teori matematika yang ada. Pengkoreksian dapat dilakukan dengan cara mengurangi nilai yang salah dengan *offset* yang ada sehingga didapatkan hasil yang benar.

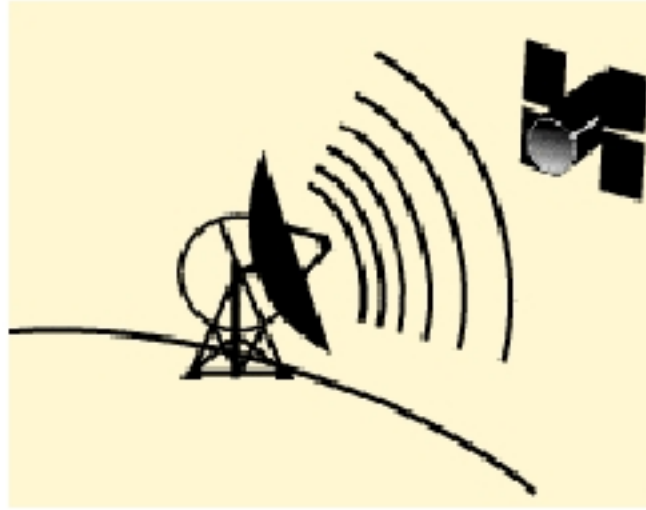
2.1.5 Menentukan posisi satelit di angkasa

Dalam pembahasan sebelumnya telah diketahui bagaimana posisi seseorang di atas bumi ini dapat dicari, yaitu dengan mencari titik potong – titik potong lingkaran yang dibuat oleh satelit-satelit GPS terhadap posisi orang itu berada. Namun dalam semuanya itu akan timbul pertanyaan, “Bagaimana dapat dicari titik potong satelit-satelit tersebut apabila tidak diketahui posisi satelit-satelit tersebut?”. Pertanyaan itu tepat sekali karena dengan hanya mengetahui jarak seseorang dengan satelit-satelit GPS maka hal itu akan menjadi sesuatu yang percuma karena tidak akan diperoleh posisi orang itu hanya dengan data-data tersebut. Jawaban atas pertanyaan itu dapat dijelaskan pada paragraf-paragraf dibawah ini.

Satelit GPS untuk pertama kali diluncurkan oleh Departemen Pertahanan Amerika Serikat pada bulan Maret 1994 sebanyak 24 satelit ke dalam orbitnya masing-masing. Jarak orbit satelit-satelit tersebut adalah 11.000 mil dari bumi dan telah diatur sedemikian rupa sehingga dalam posisi manapun di bumi, minimal akan terlihat 5 buah satelit.

Amerika menempatkan satelit-satelit tersebut pada suatu koordinat. Dimana posisi dari satelit-satelit tersebut telah diprogramkan pada setiap penerima GPS di bumi. Sehingga bila mengetahui nomor serial dari satelit yang tertangkap dapat diketahui juga posisi dari satelit-satelit tersebut. Dengan demikian posisi penerima dapat semakin nyata dan dapat dicari.

Selain itu, untuk menjamin kebenaran informasi yang diterima pemerintah Amerika juga melakukan pengecekan secara terus menerus terhadap posisi satelit tersebut dari stasiun-stasiun pemantau di bumi.



Gambar 2.13 Pemantauan Satelit Oleh Stasiun Bumi

Dengan menggunakan radar yang sangat teliti maka bujur, lintang dan ketinggian satelit-satelit tersebut dapat selalu dipantau. Kesalahan atau penyimpangan-penyimpangan yang terjadi dalam sistem ini disebut *ephemeris error*, karena hal itu akan mempengaruhi posisi orbit dari satelit-satelit tersebut.

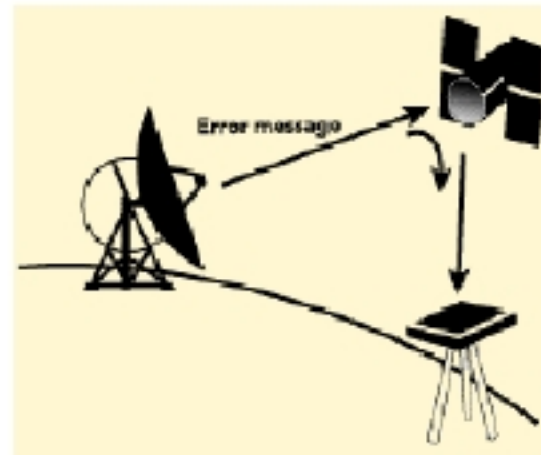
Penyimpangan orbit satelit GPS dapat terjadi karena beberapa faktor antara lain :

- Adanya gaya tarik yang dimiliki oleh bulan dan bumi.
- Adanya radiasi dari matahari yang akhirnya menyebabkan satelit-satelit GPS ini menyimpang sedikit dari orbitnya.

Sebenarnya penyimpangan ini tidak terlalu besar, tetapi jika menginginkan posisi yang tepat sekali maka faktor *error* tersebut harus diperhitungkan.

Setelah faktor kesalahan itu diketahui dan dihitung maka stasiun pemantau di bumi harus memberitahu tiap-tiap GPS *receiver* di bumi sehingga mereka dapat mengoreksi sinyal informasi yang didapat. Cara paling mudah yang dapat dilakukan oleh stasiun pemantau, yaitu mengirimkan kembali informasi *error* itu

kepada satelit GPS dan kemudian informasi itu dikirimkan secara bersama ke bumi dan diterima oleh tiap-tiap *receiver* di bumi seperti pada gambar 2.14.



Gambar 2.14 Pengiriman Kembali Informasi Error ke Satelit GPS

Dengan demikian, satelit GPS tidak lagi semata-mata merupakan suatu konstruksi yang memancarkan sinyal *pseudorandom* untuk perhitungan waktu yang tepat saja melainkan juga memancarkan sinyal-sinyal informasi navigasi dan *error-error* yang terjadi selama satelit berada di orbitnya.

2.1.6 Error dalam GPS

Dalam pembahasan di atas telah dipelajari cara kerja GPS dan ternyata sistem GPS kelihatan sangat sederhana. Hal itu memang benar secara teoritis yaitu apabila segalanya berjalan secara sempurna tanpa ada hambatan dan gangguan. Selain itu juga telah diketahui bahwa untuk mengukur jarak penerima dengan suatu satelit GPS dapat dilakukan dengan persamaan 2.1. Rumus itu hanya akan berlaku jika sinyal GPS berjalan melewati suatu ruang hampa udara sehingga tidak ada hambatan sama sekali.

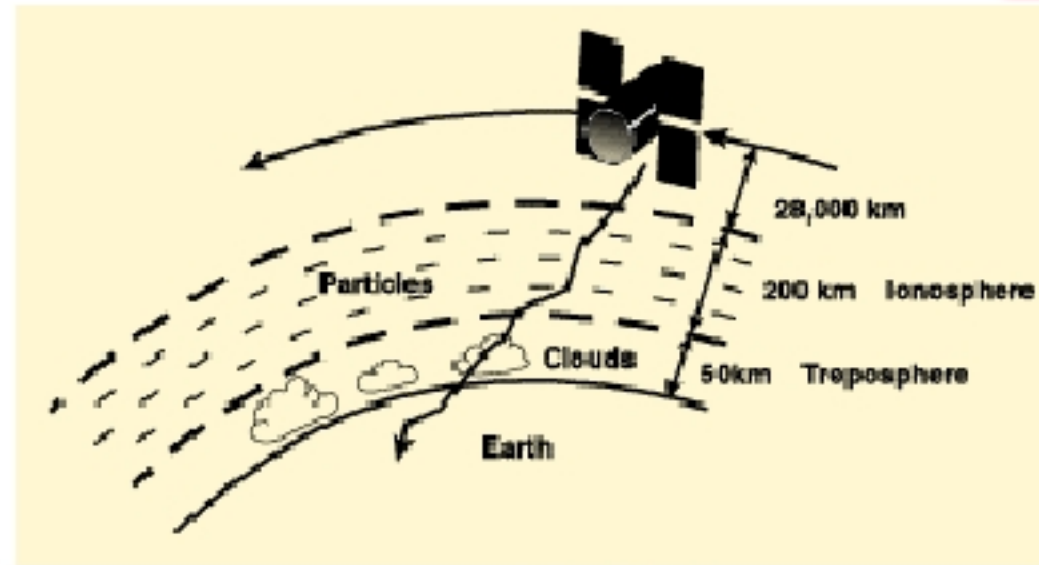
Sedangkan dalam kenyataannya hal itu tidak demikian karena untuk mengirimkan sinyal tersebut, sinyal harus melewati lapisan *atmosfir* yang berlapis-lapis, misalnya sinyal harus melewati lapisan *troposfer* yang banyak

mengandung uap air. Hal ini tentu saja akan menghambat perjalanan sinyal, akibatnya kesalahan pun terjadi. *Error* yang lain dapat disebabkan juga oleh keadaan alam pada lapisan *ionosfer* yang berada pada ketinggian 50 sampai 500 km dari permukaan bumi, dan mengandung banyak sekali partikel yang terionisasi.

Karena yang dikandung adalah partikel-partikel bermuatan maka lapisan ini akan menjadi masalah yang cukup besar bagi sinyal GPS, karena lapisan ini dapat menyebabkan sinyal GPS berbelok-belok tidak menentu yang ditimbulkan oleh medan magnet dari lapisan *ionosfer*. *Error* lain bisa disebabkan juga oleh lapisan *troposfer* yang merupakan lapisan paling bawah dari *atmosfer* bumi. Lapisan ini yang bersentuhan langsung dengan permukaan bumi. Pada lapisan ini semua peristiwa cuaca terjadi seperti awan, hujan, badai, angin dan lain sebagainya. Oleh karena itu lapisan ini banyak sekali mengandung uap air. Dengan adanya uap air dan awan maka perjalanan sinyal GPS menjadi lebih terhambat lagi. Akibatnya *delay* yang terjadi akan semakin besar. Penyebab *error* lainnya disebut dengan *multipath error*, yakni *error* yang disebabkan karena pantulan-pantulan sinyal GPS dengan benda-benda yang ada di bumi sehingga penerima menjadi sulit untuk membedakan mana sinyal yang datang secara langsung dari satelit ke *receiver* dengan sinyal yang datang ke *receiver* melalui pantulan seperti pada gambar 2.15. Hal ini mirip dengan efek *ghosting* pada sistem televisi.

Geometric Delusion of Point (GDOP) merupakan suatu faktor yang dapat menyebabkan kesalahan sebagai akibat dari struktur *geometri* yang terjadi antar satelit-satelit GPS terhadap pesawat penerima di bumi. *Error* yang lain dapat juga

terjadi karena kesalahan pada satelit sendiri ataupun kemampuan dari satelit yakni *Selective Availability* (SA) yang sengaja diciptakan oleh Amerika agar pihak-pihak lain tidak dapat memanfaatkan teknologi ini.



Gambar 2.15 Error GPS Karena Lapisan Atmosfer

2.1.7 Protokol Komunikasi Garmin *Text Out*

Kehadiran GPS dalam dunia navigasi terus berkembang sejalan dengan perkembangan teknologi. Dengan adanya teknologi *mikroprosesor* dan otomatisasi peralatan semakin berkembang dan diperbaharui setiap saat. Demikian pula dengan *GPS receiver*, yang merupakan suatu alat yang dapat digunakan sebagai alat sistem navigasi. Dengan demikian muncul suatu ide, yaitu menggabungkan peralatan ini dengan peralatan-peralatan lain, sebagai contoh *auto pilot* pada pesawat terbang. Dengan menggunakan *autopilot* maka pesawat yang dijalankan sudah tidak perlu lagi dikemudikan oleh seorang pilot.

Dengan beraneka ragamnya peralatan-peralatan yang dapat dihubungkan dengan sebuah *GPS receiver* maka harus dibentuk suatu protokol komunikasi yang *standart* sehingga suatu *GPS receiver* dapat berhubungan dengan peralatan-

peralatan yang lain, dan suatu peralatan dapat berhubungan dengan berbagai macam GPS *receiver* yang ada di pasaran. Salah satunya dikembangkan oleh Garmin dimana GPS *receiver* ini mampu memberikan *output* berupa *text out* dimana didalamnya terdapat beberapa informasi yang diperlukan.

Sistem ini bekerja pada 4800 bps dengan sistem pengiriman secara data paket. Selain itu semua data yang dikirimkan sudah bukan berupa data *byte* langsung melainkan berupa kode-kode ASCII teks yang dapat dicetak. Berikut adalah *format* dari protokol komunikasi ini :

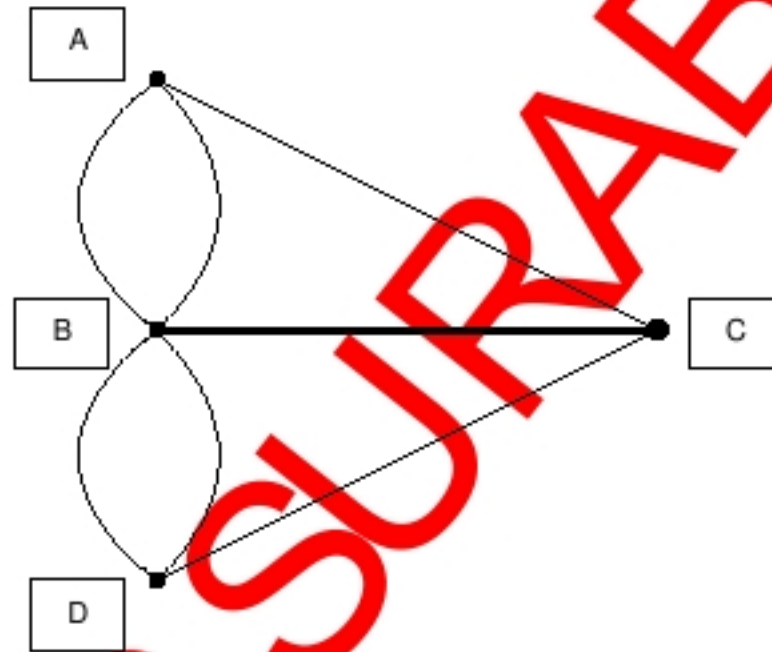
@050615081410S0715340E11242672g044+00053E0000N0000U0000

Dari *format* diatas dapat dilihat bahwa informasi sebuah lokasi diwakilkan dalam pasangan data *South* (S) atau lintang utara dan *East* (E) atau bujur timur. Sehingga dapat kita diterjemahkan sebagai sebuah lokasi yang berada pada 07°15.340 lintang utara dan 112 ° 42.672 bujur timur. (Global Positioning System Overview, 2006)

2.2 Graf

2.2.1 Teori Graf

Tulisan pertama tentang teori graf berawal pada 1736 karya Leonhard Euler. Tulisan tersebut menyajikan sebuah teori umum yang menyertakan sebuah solusi yang sekarang disebut masalah jembatan Konigsberg, seperti yang diperlihatkan dalam gambar 2.16.



Gambar 2.16 Model graf dari jembatan Koningsberg

Masalahnya adalah untuk memulai dari sembarang lokasi A, B, C, atau D menyeberangi setiap jembatan satu kali kemudian kembali lagi pada tempat memulai. *Verteks-verteks* mewakili lokasi dan rusuk-rusuk mewakili jembatan. Masalah jembatan Koningsberg sekarang disederhanakan untuk mencari siklus dalam graf yang menyertakan semua *verteks* dan semua rusuk. Untuk menghormati Euler, sebuah siklus dalam sebuah graf G yang menyertakan semua *verteks* dan semua rusuk dari G disebut sebuah **siklus Euler (Euler cycle)**.

Sebuah peta yang didalamnya terdapat banyak jalan dapat dibuat modelnya sebuah graf, karena graf digambarkan titik-titik dan garis-garis. Sebuah graf G merupakan model peta dari titik-titik yang disebut **verteks** yang

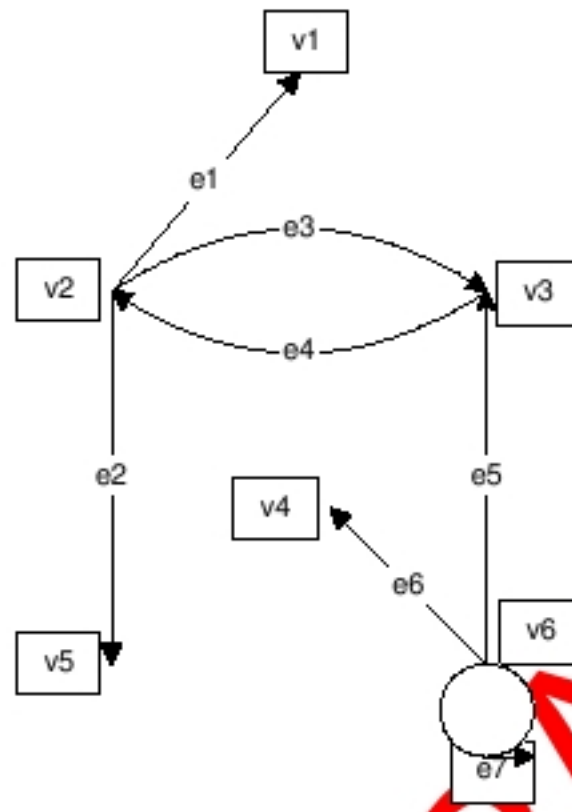
dilambangkan dengan v dan garis-garis yang menghubungkan *verteks-verteks* ini yang disebut **rusuk** (*edge*) yang dilambangkan dengan e . Jika sebuah *verteks* v_0 berjalan sepanjang sebuah rusuk ke *verteks* v_1 berjalan sepanjang rusuk yang lain ke *verteks* v_2 dan seterusnya sampai di *verteks* v_n , perjalanan tersebut disebut sebagai **lintasan** dari v_0 ke v_n .

2.2.2 Definisi Graf

Sebuah graf (atau graf tak terarah) G terdiri dari suatu himpunan V dari *verteks-verteks* (atau simpul-simpul) dan suatu himpunan E dari rusuk-rusuk (atau busur-busur) sedemikian rupa sehingga setiap rusuk $e \in E$ dikaitkan dengan pasangan *verteks* tak terurut. Jika terdapat sebuah rusuk e yang menghubungkan *verteks* v dan w , ditulis $e = (v,w)$ atau $e = (w,v)$. Dalam konteks ini (v,w) menyatakan sebuah rusuk antara v dan w dalam sebuah graf tak terarah dan bukan sebuah pasangan terurut.

Sebuah graf terarah atau di graf G terdiri dari suatu himpunan V dari *verteks-verteks* (atau simpul-simpul) dan suatu himpunan E dari rusuk-rusuk (atau busur-busur) sedemikian rupa sehingga setiap rusuk $e \in E$ menghubungkan pasangan *verteks* terurut. Jika terdapat rusuk tunggal e yang menghubungkan pasangan terurut (v,w) yang menyatakan sebuah rusuk dari v ke w .

Sebuah graf terarah diperlihatkan dalam Gambar 2.17. Rusuk terarah ditunjukkan dengan anak panah. Rusuk e_1 menghubungkan pasangan *verteks-verteks* terurut (v_1,v_2) dan rusuk e_7 menghubungkan pasangan *verteks-verteks* terurut (v_6,v_6) . Rusuk e_1 dinyatakan dalam (v_2,v_1) dan rusuk e_7 dinyatakan dalam (v_6,v_6) .



Gambar 2.17 graf terarah

Sebuah rusuk e dalam sebuah graf (tak terarah atau terarah) yang menghubungkan pasangan *verteks* v dan w dikatakan insiden pada v dan w serta v dan w dikatakan insiden pada e dan disebut **verteks-verteks damping** (*adjacent vertices*).

Jika G merupakan sebuah graf (tak terarah atau terarah) dengan *verteks-verteks* V dan rusuk-rusuk E , ditulis $G = (V, E)$. Sebuah graf dengan bilangan-bilangan pada rusuk-rusuknya disebut **graf berbobot (weighted graph)**. Jika rusuk e_1 diberi label k , maka **bobot dari rusuk** e_1 adalah k . Dalam sebuah graf berbobot, **panjang lintasan** adalah jumlah bobot rusuk-rusuk dalam lintasan.

2.2.3 Algoritma Lintasan Terpendek

Graf berbobot adalah graf dimana nilai-nilai diberikan pada rusuk-rusuknya dan panjang dari sebuah lintasan dalam graf berbobot adalah jumlah dari

bobot rusuk-rusuk, dalam lintasan tersebut. Misalkan $w(i,j)$ menyatakan bobot rusuk (i,j) . Dalam graf berbobot, sering kali dicari sebuah **lintasan terpendek** (yakni, sebuah lintasan yang mempunyai panjang minimum) antara dua *verteks* yang diketahui. Algoritma yang ditemukan E. W. Dijkstra, menyelesaikan masalah ini secara efisien.

Edsger W. Dijkstra (1930) lahir di Belanda. Dialah yang mula-mula mendukung pemrograman sebagai sebuah ilmu. Karena dedikasinya pada pemrograman maka pada saat pernikahannya di tahun 1957, ia mencantumkan profesinya sebagai seorang *programmer*. Akan tetapi, pemerintahan Belanda menyatakan bahwa profesi semacam itu tidak ada dan dia harus mengubah pada kolom profesi menjadi “fisikawan teoritis”. Ia memenangkan *Turing Award* dari *Association for Computing Machinery* pada tahun 1972. Ia juga terpilih sebagai *Schlumberger Centennial Chair in Computer Science* di Universitas Texas, Austin tahun 1984.

Algoritma Dijkstra melibatkan pemasangan label pada *verteks*. Misalkan $L(v)$ menyatakan label dari *verteks* v . Beberapa *verteks* mempunyai label sementara dan yang lainnya mempunyai label tetap

2.2.4 Algoritma Dijkstra

Algoritma ini juga sering disebut “Algoritma pemberian label”, karena menggunakan prosedur pemberian label-label pada simpul.

Algoritma djikstra digunakan untuk menentukan *shortest route* antara dua pasang simpul dalam sebuah jaringan kerja.

Catatan :

1. Algoritma ini bisa digunakan hanya pada jaringan kerja dimana bobot dari *arc* atau *edge*-nya adalah *non-negatif*.
2. Setiap simpul dalam jaringan kerja akan mempunyai label bisa *temporary* (sementara) atau *permanent* (tetap). Untuk diingat mengubah isi dari label yang sudah tetap tidak dibenarkan.
3. Sebuah label pada sebuah simpul, katakan Graph G , dalam bentuk (u,v) , dimana v menyatakan jarak Graph G dari awal melalui sebuah simple path yang mengandung *arc* (u,j) atau (edge $[u,G]$ untuk grafik tak berarah atau grafik campuran). Bila labelnya adalah *permanent*, maka vG adalah jarak terdekat dari awal ke simpul j .
4. Lintasan terpendek. Jika sebuah sisi e dalam graph G dikaitkan/dipetakan dengan sebuah bilangan real $W(e)$, maka $W(e)$ disebut bobot (*Weight*) dari e . Sebuah Graph yang setiap sisinya dikaitkan dengan bilangan real disebut Graph Bobot. Bobot dari graph G dilambangkan dengan $W(G)$, adalah jumlah bobot semua sisi G .
5. Panjang Lintasan, Panjang Lintasan dalam sebuah graph bobot adalah jumlah bobot dari dari semua sisi dalam lintasan tersebut. Jika u dan v adalah dua titik di graph G , maka lintasan (u,v) di G yang panjangnya minimum disebut lintasan terpendek.

Definisi :

1. T : adalah himpunan dari simpul-simpul yang bertanda label sementara.

2. "k" indentifikasi dari simpul yang bertanda baru saja di buat *permanent*.
3. V_0 merupakan simpul awal dari mana akan dicari jarak yang terdekat.
4. V_i merupakan simpul tujuan.

Algoritma Djikstra :

- INPUT : Graph bobot G dengan $s, t \in V(G)$.
- STEP 1 : Label titik s dengan $\lambda(s)=0$ dan untuk setiap titik v di G selain s , label v dengan $\lambda(v)=\infty$. Tulis $T=V(G)$.
- STEP 2 : misalkan $u \in T$ dengan $\lambda(u)$ minimum.
- STEP 3 : jika $u = t$ (titik terakhir), berarti STOP, dan beri pesan : panjang lintasan terpendek dari s ke t adalah $\lambda(t)$.
- STEP 4 : untuk semua sisi $e=uv, v \in T$ diganti label v dengan $\lambda(v) =$ minimum.
- STEP 5 : tulis $T = T - \{u\}$ dan selanjutnya pergi ke STEP 2.

Ada beberapa teori dalam Algoritma Djikstra

Teori 1 :

Dalam Algoritma Djikstra, jika nilai $\lambda(v)$ berhingga untuk suatu titik di graph G , maka terdapat lintasan dari s ke v di G dengan panjang $\lambda(v)$.

Teori 2 :

Dalam Algoritma Djikstra, jika titik u di graph G sudah di label *permanent*.

Dalam $\lambda(u)$, maka $\delta(u) = \lambda(u)$. (Dewiyani, 2002)

2.3 Komunikasi Data Serial

Komunikasi data berarti pengiriman data antara dua komputer yaitu antara sebuah komputer dengan terminal, atau antara terminal dengan terminal yang lain.

Komunikasi data dapat dilakukan dengan dua cara yaitu paralel dan serial. Dalam *transfer* data paralel, sering 8 atau lebih jalur (kabel konduktor) digunakan untuk men-transfer data ke suatu *device* yang berjarak hanya beberapa kaki. Contoh *transfer* paralel adalah printer dan *hard disk* yang menggunakan kabel dengan banyak jalur. Meskipun dalam kasus-kasus seperti ini banyak data bisa di *transfer* dalam waktu singkat dengan menggunakan banyak kabel yang disusun paralel, tetapi jaraknya tidak bisa jauh dan biaya relatif lebih mahal. Untuk men-transfer data ke suatu *device* yang terletak lebih jauh maka, digunakan metode serial. Dalam komunikasi serial, data dikirim satu *bit* dalam suatu waktu, berbeda dengan komunikasi paralel, dimana data dikirim satu *byte* atau lebih dalam suatu waktu. Kelebihan metode serial ini adalah selain ia dapat digunakan dalam jarak yang jauh, ia juga memerlukan biaya yang lebih murah jika dibandingkan dengan metode paralel.

Komunikasi data serial menggunakan dua metode, *asinkronous* dan *sinkronous*. Metode *sinkronous* men-transfer suatu blok data (karakter) pada suatu waktu sedangkan *asinkronous* men-transfer suatu *byte* tunggal pada suatu waktu.

Adalah mungkin untuk membuat *software* untuk digunakan dengan metode-metode di atas, tetapi programnya bisa membosankan dan panjang. Karena itu, ada chip-chip khusus yang dibuat oleh banyak pabrik untuk

komunikasi data serial. Chip-chip ini secara umum dikenal sebagai *Universal Asynchronous Receiver-Transmitter* (UART) dan *Universal Synchronous-Asynchronous Receiver-Transmitter* (USART). Pada komputer IBM PC *Compatible port* serial atau sering dikenal sebagai COM menggunakan standart UART 8250.

2.3.1 Transmisi *simplex*, *half-duplex* dan *full-duplex*

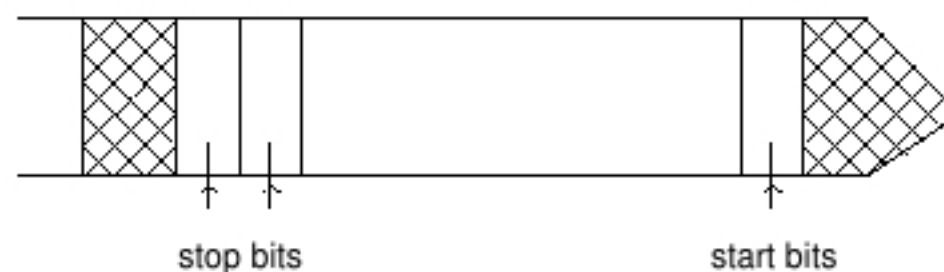
Transmisi *simplex* berarti 1 arah saja, dimana 1 titik selalu bertindak sebagai *transmitter* (pengirim), sedangkan titik lain selalu *receiver* (penerima). Pola ini sangat sederhana sebagai contoh adalah *printer*.

Tranmisi *half – duplex* komunikasi dilakukan dua arah tetapi secara bergantian, karena hanya ada 1 jalur komunikasi yang terpakai. Sebagai contoh komunikasi lewat radio amatir seperti CB

Komunikasi *full – duplex* dilakukan dua arah dalam waktu bersamaan/secara simultan. *Full duplex* membutuhkan dua kabel konduktor (selain *ground*), satu untuk transmisi dan satu untuk penerimaan, agar bisa men-transfer dan menerima data secara sekaligus. Sebagai contoh adalah komunikasi lewat telepon

2.3.2 Komunikasi serial *asinkronous* dan *sinkronous*

Data yang dikirim dengan *asinkronous* adalah sebagai berikut :



Gambar 2.18 Metode *asinkronous*

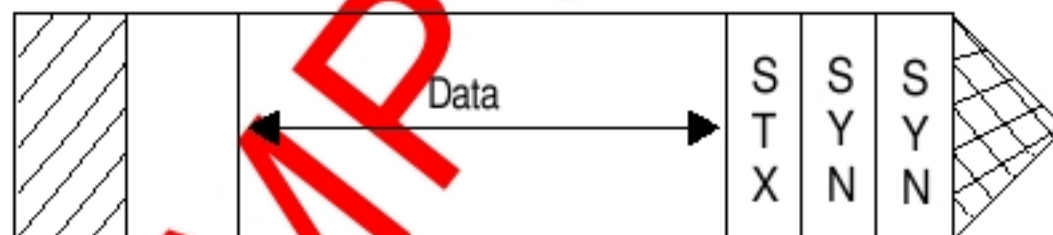
Karakter diawali dengan *start bit* dan diakhiri dengan bit. *Bit start* disertakan demi sinkronisasi. Pada saat penerimaan *bit start* memeriksa apakah sinkronisasi *bit* dan sinkronisasi karakter benar-benar terlaksana.

Keuntungan metode *asinkronous* adalah cara pengiriman data yang sederhana dan tidak diperlukan jalur tambahan untuk menyamakan *clock* pada *transmitter* dan *receiver*.

Kerugiannya adalah efisien pengiriman yang turun karena untuk 8 *bit* data, minimum ada 2 *bit* data (*start/stop bit*) yang tidak mengandung informasi.

Dengan kata lain *over head asinkronous* minimum 20%. Kelemahan lain dari *asinkronous* adalah *band rate* yang relatif rendah. *Band rate* asinkron adalah 300, 1200, 2400, 4800, 9600, 19200 *bit per second*.

Data dikirim dengan *sinkronous* sebagai berikut ;

Gambar 2.19 Metode *sinkronous*

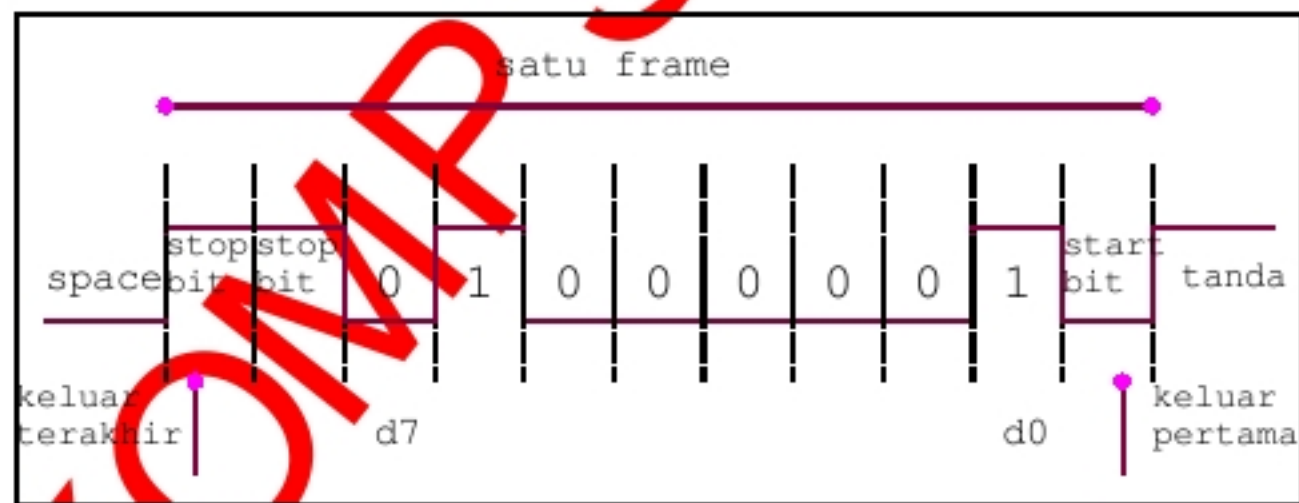
Setiap kali *receiver* menerima karakter SYN, berupa 1 atau 2 *byte* dengan kode tertentu, maka *receiver* akan menyamakan *clock*. Kemudian 1 blok data (*multi bytes*) diterima dan kebenarannya dicocokkan dengan blok *check character* sebelum *end of Text* (ETX).

Dengan cara ini, maka *over head* transmisinya lebih rendah dibandingkan dengan *asinkronous* sehingga cara ini lebih efisien.

Kecepatan transmisi *sinkronous* lebih tinggi dari *asinkronous*.

2.3.3 Start bit dan stop bit

Komunikasi data serial *asinkronous* digunakan secara luas untuk transmisi berorientasi-karakter, dan *transfer* data berorientasi-blok menggunakan metode *sinkronous*. Dalam metode *asinkronous*, setiap karakter diletakkan antara *start* dan *stop bit* yang disebut *framing*. *Start* dan *stop bit* mengirimkan data *framing* untuk komunikasi *asinkronous* data seperti ASCII karakter, dipaketkan di antara sebuah *start bit* dan sebuah *stop bit*. *Start bit* terdiri dari satu *bit* tetapi *stop bit* bisa satu atau dua *bit*. *Start bit* selalu '0' (*low*) dan *stop bit* adalah '1' (*high*). Sebagai contoh, lihat Gambar 2.20 dimana karakter ASCII "A", kode binernya 0100 0001, ditunjukkan di antara sebuah *start bit* dan sebuah *single stop bit*. Dalam hal ini LSB dikirim terlebih dahulu.



Gambar 2.20 Framing ASCII "A" (41H)

Pada Gambar 2.20, bila tidak ada *transfer*, sinyalnya bernilai 1 (*high*), yang disebut sebagai *mark* dan bernilai 0 (*low*) disebut sebagai *space*. Perhatikan bahwa transmisi mulai dengan *start bit* yang diikuti oleh D0 (*LSB*), kemudian sisa

bit-bit sampai MSB (D7), dan akhirnya, satu *stop bit* yang menunjukkan akhir dari karakter "A".

Dalam komunikasi serial *asinkronous*, chip-chip *peripheral* dan modem dapat di program untuk data selebar 7 atau 8 *bit*. Ini sebagai tambahan dari jumlah *stop bit*, 1 atau 2. Sementara dalam sistem yang lebih lama karakter-karakter ASCII adalah 7-bit, karena adanya karakter *extended ASCII*, *bit* ke 8 menjadi *common*.

Dalam beberapa sistem lama, disebabkan kelambatan peralatan mekanik yang menerima, 2 *stop bit* digunakan untuk memberikan peralatan tersebut cukup waktu untuk mengorganisasi dirinya sendiri sebelum transmisi dari *byte* berikutnya. Tetapi, dalam PC-PC *modern* penggunaan 1 *stop bit* adalah umum. Dengan asumsi bahwa kita men-transfer file *text* dari karakter ASCII menggunakan 2 *stop bit*, kita punya total 11 *bit* untuk tiap karakter yaitu 8 *bit* untuk kode ASCII-nya, dan 1 dan 2 *bit* masing-masing untuk *start* dan *stop bit*. Oleh karena itu, untuk setiap karakter 8-bit ada 3 *bit* ekstra, atau lebih dari 30%.

Dalam beberapa sistem untuk menangani integritas data, *bit parity* dari *byte* karakter dimasukkan dalam *data frame*. Ini berarti bahwa untuk tiap karakter (7- atau 8-bit, tergantung pada sistem) kita punya *bit parity* tunggal sebagai tambahan dari *start bit* dan *stop bit*. *Bit parity* adalah ganjil atau genap. Dalam kasus *bit parity*-ganjil jumlah bit data, termasuk *bit parity*, punya *number* ganjil dari 1-an. Serupa dengan itu, dalam sebuah *bit parity*-genap jumlah *total bit*, termasuk *bit parity*, adalah genap. Sebagai contoh, karakter ASCII "A", biner 0100 0001, punya 0 untuk bit *parity*-genap. Chip-chip UART memungkinkan pemrograman *bit parity*-ganjil, *parity*-genap, dan pilihan tanpa-*parity*,

sebagaimana yang akan kita lihat dalam pembahasan berikutnya. Jika suatu sistem membutuhkan *parity*, *bit parity* ditransmisikan setelah MSB, dan diikuti oleh *stop bit*.

2.3.4 Laju transfer data

Laju transfer data dalam komunikasi data serial dinyatakan dalam *bits per second (bps)*. Istilah lain yang digunakan untuk *bps* adalah *baud rate*. Bagaimanapun, *baud rate* dan *bps rate* tidaklah benar-benar sama. Ini disebabkan fakta bahwa *baud rate* adalah istilah modem dan didefinisikan sebagai jumlah perubahan sinyal per detik. Dalam modem, ada kejadian-kejadian ketika suatu perubahan tunggal dari sinyal men-transfer beberapa bit data. Sejauh kabel konduktor yang menjadi fokus bahasan, *baud rate* dan *bps* adalah sama, dan dengan alasan ini dalam hal ini kita bisa saling mempertukarkan *term bps* dan *baud rate*.

Laju sistem data dari suatu sistem komputer tergantung pada *port-port* komunikasi yang ada dalam sistem itu. Sebagai contoh, IBM PC/XT awal bisa mentransfer data pada laju 100 sampai 9600 bps. Tetapi dalam tahun-tahun terakhir, PC, PS, dan komputer kompatibel 80x86 men-transfer data pada laju sebesar 19.200 *bps*. Harus dicatat bahwa dalam komunikasi data serial *asinkronous*, *baud rate* biasanya dibatasi sampai 100.000 *bps*.

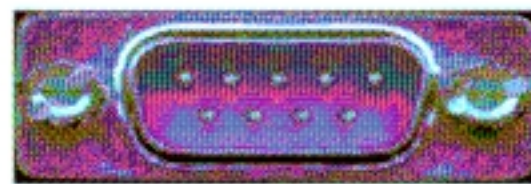
2.3.5 RS232 dan Standart-Standart I/O Serial Lain

Untuk memungkinkan kompatibilitas di antara peralatan komunikasi data yang dibuat oleh berbagai pabrik, *standart interfacing* yang disebut RS232 telah

ditetapkan oleh *Electronics Industries Association* (EIA) pada tahun 1960. Pada tahun 1963 *standart* itu dikembangkan dan dikenal sebagai RS232A. RS232B dan RS232C masing-masing diperkenalkan pada 1965 dan 1969. Dalam buku ini secara sederhana kita menyebutnya sebagai RS232. Dewasa ini, RS232 adalah *standart interfacing I/O serial* yang paling banyak digunakan. Tetapi, karena *standart* itu ditetapkan lama sebelum kemunculan keluarga TTL *logic*, tingkat tegangan *input* dan *output*-nya tidak kompatibel TTL. Dalam RS232, '1' direpresentasikan dengan -3 s/d -25 V, sedangkan bit 0 adalah +3 s/d +25 V, sehingga -3 sampai +3 tidak didefinisikan.

RS232 sebagai komunikasi serial mempunyai 9 pin yang memiliki fungsi masing-masing. Pin yang biasa digunakan adalah pin 2 sebagai *received data*, pin 3 sebagai *transmitted data*, dan pin 5 sebagai *ground signal*. Karakteristik elektrik dari RS232 adalah sebagai berikut :

- *Space (logic 0)* mempunyai level tegangan sebesar +3 s/d +25 Volt.
- *Mark (logic 1)* mempunyai level tegangan sebesar -3 s/d -25 Volt.
- Level tegangan antara +3 s/d -3 Volt tidak terdefiniskan.
- Arus yang melalui rangkaian tidak boleh melebihi dari 500 mA., ini dibutuhkan agar sistem yang dibangun bekerja dengan akurat.



Gambar 2.21. RS232 (Sebagai Komunikasi Serial)

Tabel 2.1 Fungsi masing-masing pin RS232

RS232 Pin Assignments (DB9 PC signal set)	
Pin 1	<i>Received Line Signal</i> Detector (Data Carrier Detect)
Pin 2	<i>Received Data</i>
Pin 3	Transmit Data
Pin 4	Data Terminal Ready
Pin 5	Signal Ground
Pin 6	Data Set Ready
Pin 7	Request To Send
Pin 8	Clear To Send
Pin 9	Ring Indicator