

# PENTINGNYA SINKRONISASI WAKTU PADA JARINGAN KOMPUTER

Iswanto

Program Studi Informatika Universitas Nurtanio Bandung  
Jl. Pajajaran 219 Husein Sastranegara Bandung  
email: isw789ng@gmail.com

---

## ABSTRAK

Waktu yang sinkron (selaras) pada semua peralatan adalah hal yang sangat penting di dalam suatu sistem yang menggunakan peralatan jaringan karena setiap aspek perencanaan (*planning*), pengelolaan (*managing*), pengamanan (*security*) dan *debuging* membutuhkan parameter waktu yang akurat sebagai acuan dalam mengeksekusi setiap proses. Sistem data center, wired dan wireless communications, financial exchanges, industrial networks dan sistem komunikasi lainnya memerlukan waktu yang akurat dan sinkron untuk menjamin ketepatan data, eksekusi proses dan kemanan.

Untuk menjamin sinkronisasi waktu pada semua peralatan tidak bisa mengandalkan pada jam komputer/peralatan (*computer clock/ device or equipment clock*) yang dimiliki oleh masing-masing peralatan, karena *computer clock* dikelola oleh baterai dan *clock calender* yang bisa menghasilkan penyimpangan maksimum 1 detik per hari. Permasalahan ini bisa diatasi dengan menggunakan sebuah *time server* (server waktu) yang berfungsi sebagai acuan waktu untuk semua peralatan, yang secara periodik akan mmeberikan *update* waktu ke semua peralatan yang tergabung dalam sistem.

Microsoft dan Linux telah menyediakan NTP Server dan NTP Client yang mampu memberikan layanan sinkronisasi waktu, tetapi penggunaannya kadang-kadang terjadi permasalahan karena perbedaan platform, versi sistem operasi dan masalah konfigurasi dan setting lainnya.

Pada penelitian ini dibangun sebuah rancangan awal aplikasi client-server yang berfungsi untuk sinkronisasi waktu untuk tujuan khusus, bisa dioperasikan tanpa tergantung dengan layanan bawaan dari sistem operasi.

**Kata kunci:** sinkronisasi waktu, server waktu

---

## 1. PENDAHULUAN

Sinkronisasi waktu (*time synchronization*) adalah salah satu topik dalam ilmu komputer yang bertujuan untuk menyelaraskan (*synchronize*) waktu yang berjalan secara *independen* pada setiap elemen sistem dari sebuah sistem yang tergabung dalam sebuah jaringan (*network*). Karena jika tidak ada upaya sinkronisasi waktu pada keseluruhan elemen sistem yang ada, walaupun pada awalnya waktu sudah ditetapkan secara akurat, *clock time* pada masing-masing elemen sistem akan berbeda setelah beberapa waktu berjalan. Karena *clock time* dihitung oleh *computer clock* yang dikelola oleh baterai dan *clock calender* yang bisa menghasilkan penyimpangan maksimum 1 detik per hari. Masing-masing elemen sistem akan menghitung *clock time* secara independen dengan tingkat keakuratan yang berbeda-beda sehingga semakin lama perbedaan akan terakumulasi menjadi perbedaan waktu yang signifikan.

Perkembangan teknologi sangat sulit di jauhkan dari kehidupan manusia, khususnya bermunculannya *smartphone* yang membantu pekerjaan manusia atau sekedar mengenalkan gaya hidup baru. *Smartphone* yang paling banyak diminati di pasaran adalah *smartphone* berbasis *android*. Banyak keuntungan yang bisa didapatkan dari android. Salah satunya adalah *android* memiliki banyak *game*, baik yang gratis maupun berbayar yang tersedia di *market android*.

Pada sistem jaringan komputer moderen yang melibatkan banyak peralatan (*device*), sinkronisasi waktu merupakan hal yang kritical dalam mendukung aspek perencanaan (*planning*), pengelolaan (*managing*), pengamanan (*security*) proses dan data. Sinkronisasi waktu sangat penting digunakan pada wired dan wireless data communications, financial exchanges, industrial networks, dan sistem komunikasi lainnya yang memerlukan

parameter waktu yang akurat untuk menjamin ketepatan data, eksekusi proses dan keamanan.

Sinkronisasi waktu sangat diperlukan pada beberapa aplikasi berikut ini :

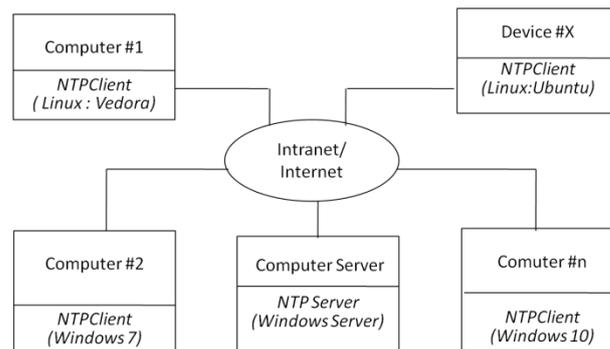
- Komunikasi radio yang menggunakan akses media *Time Division Multiple Access (TDMA)*.
- Untuk memelihara konsistensi filesystem yang di-*share* pada jaringan dari proses modifikasi.
- Pemakaian pada *billing services* yang memerlukan parameter waktu yang akurat.
- Beberapa *financial services* yang membutuhkan *highly accurate time keeping* yang dipesyaratkan oleh hukum.

Sinkronisasi waktu bisa dilakukan dengan menggunakan sebuah *time server* (server waktu) yang berfungsi sebagai acuan waktu untuk semua peralatan, yang secara periodik akan memberikan *update* waktu dengan ketelitian tertentu ke semua peralatan yang tergabung dalam sistem. Microsoft dan Linux telah menyediakan NTP Server dan NTP Client yang mampu memberikan layanan sinkronisasi waktu, tetapi penggunaannya kadang-kadang terjadi permasalahan karena perbedaan platform, versi sistem operasi, kelengkapan komponen, masalah konfigurasi dan setting lainnya.

Pada penelitian ini dibangun sebuah rancangan awal aplikasi *client-server* yang berfungsi untuk sinkronisasi waktu untuk tujuan khusus, bisa dioperasikan tanpa tergantung dengan layanan bawaan dari sistem operasi.

## 2. NETWORK TIME PROTOCOL

Sebagai gambaran bagaimana cara melakukan sinkronisasi waktu, pada bab ini dijelaskan sekilas tentang penggunaan NTP yaitu sebuah protokol telah digunakan oleh Microsoft dan Linux. NTP adalah singkatan dari *Network Time Protocol*, adalah sebuah protocol untuk meng-sinkronkan waktu sistem (*clock*) pada komputer terhadap sumber yang akurat, melalui jaringan intranet ataupun internet. NTP Server mentransmisikan paket informasi waktu kepada komputer client melalui NTP Client yang melakukan *request*. Ilustrasi penggunaan NTP ditunjukkan pada Gambar 2.1. NTP Server sangat bermanfaat sekali untuk mengelola jaringan yang sangat ketat sekali dalam urusan waktu.



Gambar 2.1 Network Time Protocol

NTP menggunakan *port* komunikasi *UDP* dengan nomor 123. Protokol ini didesain untuk dapat bekerja dengan baik meskipun media komunikasinya bervariasi, mulai dari yang waktu latensinya tinggi hingga yang rendah, mulai dari media kabel sampai dengan media udara. Protokol ini memungkinkan perangkat-perangkat komputer dalam jaringan tetap dapat melakukan sinkronisasi waktu dengan sangat tepat dalam berbagai media tersebut.

Sebuah jaringan NTP biasanya mendapatkan perhitungan waktunya dari sumber waktu yang terpercaya seperti misalnya *radio clock* atau *atomic clock* yang terhubung dengan sebuah *time server*. Kemudian jaringan NTP ini akan mendistribusikan perhitungan waktu akurat ini ke dalam jaringan lain. Sebuah NTP client akan melakukan sinkronisasi dengan NTP server dalam sebuah interval *pooling* yang biasanya berkisar antara 64 sampai 1024 detik. Namun, waktu sinkronisasi ini bisa berubah secara dramatis bergantung kepada kondisi dan keadaan jaringan yang akan digunakannya.

Untuk mengaplikasikan NTP, perlu dilakukan *setting* atau pengaturan baik pada sisi server maupun pada sisi client. Pengaturan yang perlu dilakukan pada sisi server :

- *Enable Windows Time Services*
- *Enabe NTP Server*
- *Running NTP Server*

Sedangkan pengaturan pada sisi client adalah :

- *Enable Windows Time Service*
- *Enabe NTP Client*
- *Menentukan IP address NTP Server*
- *Running NTP Client*

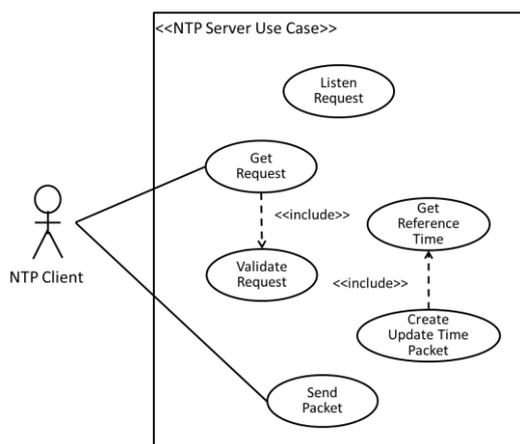
Akurasi dari sinkronisasi waktu tergantung pada *time server* yang digunakan. *Time Server*

adalah komputer server yang membaca waktu aktual dari jam referensi dan mendistribusikan informasi ini kepada kliennya menggunakan jaringan komputer. Server waktu dapat berupa server waktu jaringan lokal atau server waktu internet.

Referensi waktu yang digunakan oleh *Time Server* dapat berupa server waktu lain di jaringan atau internet, jam radio yang terhubung, atau jam atom. Sumber waktu sebenarnya yang paling umum adalah GPS atau jam master GPS. Tugas dari *Time Server* adalah menyediakan waktu yang akurat. Server jaringan yang ada (misalnya server file) dapat menjadi server waktu dengan perangkat lunak tambahan.

### 3. SPESIFIKASI TIME SYNCHRONIZER

Spesifikasi fungsional aplikasi *Time Synchronizer* yang akan dibangun direpresentasikan dalam bentuk diagram *use case* pada gambar berikut ini :



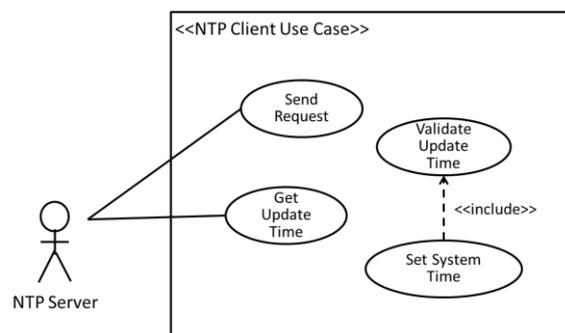
Gambar 3.1 Diagram Use Case NTP Server

Server NTP mempunyai lima fungsi utama yaitu :

- (1) *Listen Request* yaitu Server selalu menunggu *request* dari setiap client yang secara periodik mengirimkan permintaan *time\_update* untuk menyelaraskan waktunya.
- (2) *Berhasil* melakukan *Get Request* ketika ada client yang mengirimkannya melalui *socket* dan melakukan validasi terhadap *request* tersebut.
- (3) Melakukan *Get Reference Time* dari waktu sistem yang digunakan untuk

menyelaraskan (sinkronisasi) waktu pada seluruh client yang tergabung dalam LAN.

- (4) *Reference Time* disusun dalam bentuk paket data dengan melakukan *Create Update Time* sebelum dikirimkan ke client.
- (5) *Send Packet* kepada client melalui *socket* dengan port yang telah disepakati.



Gambar 3.2 Diagram Use Case NTP Client

Pada sisi client terdapat empat fungsi utama yaitu :

- (1) *Send Request*, mengirimkan permintaan *update time* kepada server secara periodik.
- (2) *Get Update Time*, menerima *update time* dari server.
- (3) Melakukan *Validate Update Time* untuk menentukan validitas dari *update time* yang diterima.
- (4) Jika valid, maka client melakukan *setting* waktu pada waktu sistem.

### 4. DESAIN DAN IMPLEMENTASI

Aplikasi *Time Synchronizer* adalah aplikasi yang dirancang dan dibangun untuk memudahkan melakukan sinkronisasi waktu pada komputer dan peralatan lain yang tergabung dalam sebuah LAN. Ide ini muncul ketika penulis menemui kesulitan dalam melakukan sinkronisasi waktu dalam jaringan komputer yang menggunakan beberapa platform dan versi sistem operasi. Sinkronisasi pada komputer/operating sistem tertentu versi lama tidak bisa berjalan dikarenakan sistem operasi tersebut belum dilengkapi komponen NTP.

### 4.1 DESAIN ALGORITMA

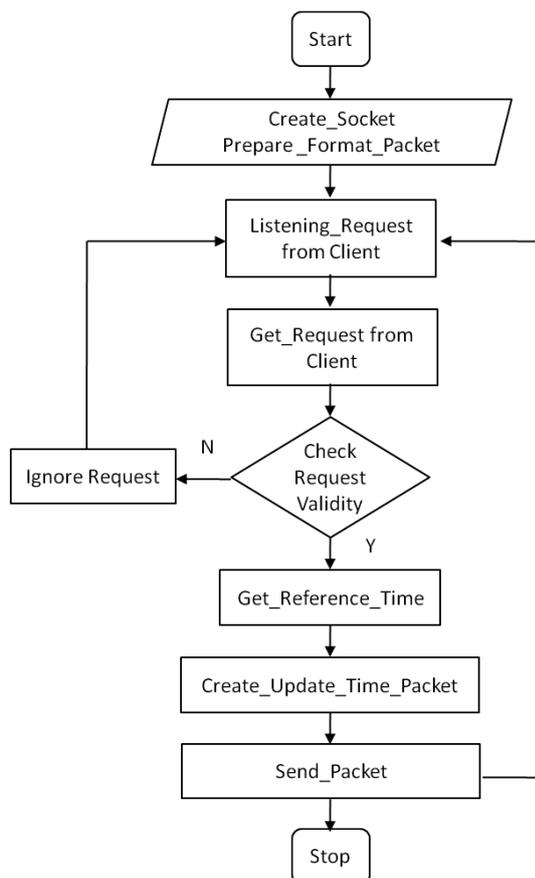
Time Synchronizer merupakan aplikasi *command line* terdiri dari aplikasi Server dan Client. Aplikasi ini menggunakan *socket UDP* sebagai saluran komunikasi data yaitu untuk mengirimkan request dari client dan update time dari server. Pengiriman update time sebagai waktu referensi melibatkan pertimbangan *client time offset* seperti ditunjukkan pada Gambar 4.1 berikut :



Gambar 4.1 Client Time Offset

*Update time* yang diterima oleh server harus dikurangi dengan *time offset* yang merupakan waktu transmisi data antara client dan server, sehingga menghasilkan besaran waktu yang tepat.

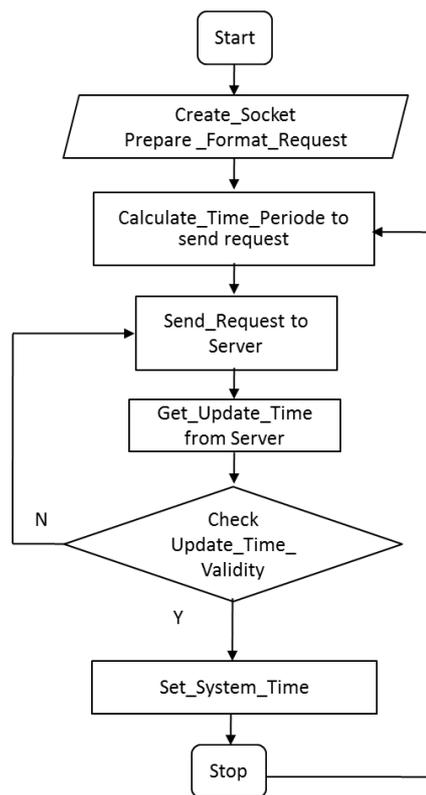
Desain algoritma secara umum ditunjukkan pada Gambar 4.2 dan 4.3



Gambar 4.2 Algoritma Server

Pada waktu Server dijalankan, server menyiapkan socket sebagai kanal komunikasi dengan client dilanjutkan menunggu *request* dari client. Proses menunggu request ini dilakukan dengan cara melakukan looping secara periodik memeriksa buffer apakah ada paket request dari client.

Jika terdapat paket dalam *buffer*, diambil kemudian diperiksa apakah paket tersebut berisi *request* dengan panjang yang sesuai dengan yang disepakati. *Request* yang valid akan diproses lebih lanjut yaitu dengan membaca waktu referensi (dalam hal ini adalah waktu dari server sebagai referensi), waktu referensi disusun dalam bentuk paket update time dan dikirimkan pada client yang melakukan *request*. Dengan cara ini maka semua komputer yang tergabung dalam LAN akan memiliki waktu yang sama.



Gambar 4.3 Client

Pada sisi client, selalu menghitung waktu/periode untuk menentukan kapan harus mengirim *request* ke Server. Setelah *request* dikirim, client akan menerima update time dari server dan memeriksa validitasnya. Jika update

time yang dikirim oleh server valid, maka client kan melakukan setting waktu sistemnya.

Skenario lain sebagai alternatif bisa dilakukan juga, server tidak perlu menunggu request dari client untuk mengirimkan update time, tetapi bisa melakukannya secara periodik, misalnya setiap menit sekali dan *membroadcast* kepada semua komputer yang tergabung dalam LAN.

Aplikasi Server dan Client didesain untuk bisa dijalankan pada komputer dengan operating sistem linux maupun windows, sehingga memudahkan bagi administrator jika di dalam LAN terdapat beberapa platform sistem operasi maupun beberapa versi dari sistem operasi.

Dengan waktu yang tersinkronisasi untuk semua komputer maka akan terjamin keakuratan dan konsistensi dari data date-time dari file, data base dan sistem yang menggunakan aplikasi jaringan yang melibatkan penjadwalan proses dan komunikasi data secara TDMA.

## 4.2 IMPLEMENTASI

Aplikasi Time Synchronizer (Server dan Client) dikembangkan menggunakan Gnu CC. Untuk bisa diaplikasikan pada platform windows, maka program harus di-compile menggunakan compiler yang mampu melakukan *porting* dari lingkungan unix/linux ke dalam lingkungan windows.

Pada penelitian ini digunakan *Cygwin* sebagai *tool* untuk *porting* dari lingkungan unix/linux ke dalam lingkungan windows. *Cygwin* adalah sebuah lingkungan Unix-like dengan antarmuka *command line* jika dijalankan pada windows. *Cygwin* menyediakan konversi program dan data dari lingkungan *Unix-like* ke aplikasi berbasis windows, data, dan sumber daya sistem lainnya.

*Cygwin* terdiri dari dua bagian yaitu sebuah *dynamic-link library (dll)* sebagai lapisan kompatibilitas API dalam bentuk C standard library, yang menyediakan sebagian besar dari API POSIX, dan koleksi program dan aplikasi yang menyediakan tampilan dalam lingkungan Unix-like.

*Cygwin* pada awalnya dikembangkan oleh Cygnus Solutions, yang kemudian diakuisisi oleh Red Hat. Perangkat lunak ini adalah software gratis dan *open source*, yang dirilis di

bawah GNU Lesser General Public License Versi 3.

Server dan Client menggunakan pustaka seperti berikut :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/time.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <arpa/inet.h>
```

Variabel penting yang digunakan adalah :

```
struct timeval tv;
```

menggunakan pustaka `sys/time.h`

Fungsi penting yang digunakan adalah :

- (1) *recvfrom (...)*  
Berfungsi untuk membaca data yang terdapat dalam buffer.
- (2) *sendto(...)*  
Berfungsi untuk mengirimkan paket data melalui socket
- (3) *getReferenceTimestamp ()*  
Berfungsi untuk membaca data waktu yang digunakan sebagai waktu referensi.
- (4) *settimeofday()*  
Berfungsi untuk melakukan setting waktu sistem.
- (5) *gettimeofday ()*  
Berfungsi untuk mendapatkan penunjukkan waktu saat ini dari sistem

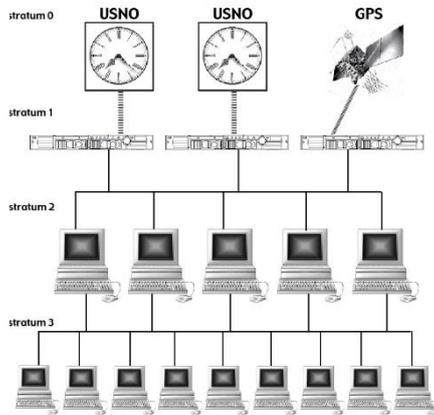
Sebagian dari *line of code* Server adalah sebagai berikut :

```
while(1) //Looping untuk listening dan
  kirim update time
{
  ...
  ...
  n = recvfrom( sockfd, ( char*) &packet,
  sizeof(ntp_packet), 0, (struct
  sockaddr*)&serv_addr, &addr_len);

  if (n != sizeof(packet))
  {
    printf("ERROR ukuran packet request
    tidak sama dengan 48 bytes\n");
  }
  ...
  ...
```



Jika sistem membutuhkan ketelitian yang cukup tinggi maka server bisa dihubungkan dengan Time Server yang lebih akurat. Penggunaan Time Server terdapat istilah "stratum" yang digunakan untuk memberi label kedekatan dengan server waktu pusat atau berkualitas tinggi.



Gambar 5.4 Ilustrasi stratum

Stratum menunjukkan tempat Time Server waktu tertentu dalam hierarki server. Skala adalah 1 hingga 15 di mana 1 adalah yang paling akurat dan kemungkinan merupakan perangkat keras fisik yang sangat khusus. Beberapa client akan menolak pembaruan waktu dari server yang stratanya terlalu tinggi, dan sebagian besar akan lebih suka sumber waktu strata rendah untuk yang lebih tinggi. Hal ini bisa menjadi perangkat bagi administrator yang mengatur server waktu *in-house* tanpa sumber waktu sebenarnya.

## 6. KESIMPULAN DAN SARAN

Deskripsi dan pembahasan pada bab-bab sebelumnya menghasilkan kesimpulan sebagai berikut :

- (1) Sinkronisasi (penyelarasan) waktu (*time synchronization*) merupakan hal yang sangat penting pada sistem yang menggunakan parametr waktu sebagai parameter utama (penting) seperti pada sistem komunikasi TDMA atau pada billing services.
- (2) Pada penelitian ini telah dibangun sebuah aplikasi Time Synchronizer (Client/Server) menggunakan bahasa C yang bisa dijalankan pada lingkungan windows maupun unix/linux.

- (3) Peningkatan ketelitian penyelarasan waktu bisa dilakukan menggunakan time server khusus.

## DAFTAR PUSTAKA

1. Suri, J.M, Nath, B.K, "Time Synchronization in IP Network", Telecommunication Center Engineering, New Delhi
2. Symmetricom, "The Importance of Network Time Synchronization", White Paper
3. <https://www.endruntechnologies.com/netw-ork-time-synchronization.htm>
4. <http://pubs.opengroup.org/onlinepubs/7908799/xsh/systime.h.html>
5. [https://en.wikipedia.org/wiki/Time\\_server](https://en.wikipedia.org/wiki/Time_server)