BAB II

LANDASAN TEORI

2.1 Transmisi Data

Transmisi adalah pergerakan informasi melalui sebuah media telekomunikasi. Transmisi memperhatikan pembuatan saluran yang dipakai untuk mengirim informasi, serta memastikan bahwa informasi sampai secara akurat dan dapat diandalkan.

Data itu sendiri merupakan bentuk jamak dari datum yang berarti informasi, jelasnya data itu dapat berupa apa saja dan dapat ditemui di mana saja, kemudian kegunaan dari data adalah sebagai bahan dasar yang obyektif (relatif) di dalam proses penyusunan kebijaksanaan dan keputusan oleh pimpinan organisasi.

Transmisi data dapat diartikan sebagai pengiriman data dari satu media ke media lainnya. Dalam skripsi ini menjelaskan transimisi data dengan menggunakan media komputer melalui media kabel. Pengiriman data menggunakan media kabel (*wire*) adalah sebagai penghubung satu komputer dengan komputer lainnya. Adapun tipe-tipe kabel yang digunakan dalam media transmisi antara lain :

2.1.1 Kabel *Unshielded Twisted Pair* (UTP)

Kabel *twisted pair* terdiri dari dua tipe yaitu *shielded and unshielded.* Unshielded Twisted Pair (UTP) adalah yang paling popular dan umumnya merupakan pilihan yang terbaik untuk jaringan sederhana.

2.1.2 Kabel *Shielded Twisted Pair* (STP)

Kekurangan kabel jenis ini adalah sangat sensitive terhadap sinyal radio dan listrik. Kabel seperti sangat baik digunakan dimana lingkungan pengaruh listrik kurang, serta biasanya digunakan pada jaringan yang menggunakan topologi *Token Ring*.

2.1.3 Kabel *Coaxial*

Kabel *coaxial* adalah kabel yang memiliki satu *Copper Conductor* dibagian tengahnya. Sebuah lapisan plastic menutupi diantara konduktor dan lapisan pengaman serat besi. Lapisan serat besi tersebut membantu menutupi gangguan dari arus listrik, lslulintas kendaraan atau mesin , dan komputer.

2.1.4 Kabel *Fibre Optic*

Kabel *fibre optic* (serat optik) mempunyai kemampuan mentransmisi sinyal melalui jarak yang relative jauh dari pada kabel *coaxial* ataupun kabel *twisted*, serta mempunyai kecepatan yang baik.

Adapun tabel perbandingan dalam segi kapasitas pengiriman, bandwith, dan jarak yang dapat dilalui oleh media kabel diatas

Tabel 2.1 Perbandingan Jenis Kabel Transmisi Data

Jenis Kabel	Maksimal Kecepatan Pengiriman	Bandwith	Jarak antar repeater
Twisted Pair	4Mbps	250MHz	2-10km
Coaxial	500Mbps	350MHz	1-10km
Fiber Optik	2Gbps	2GHz	10-100km

Pengiriman data menggunakan media tanpa kabel (*wireless*) adalah sebagai penghubung satu komputer dengan komputer lainnya. Antara lain adalah :

1. Infrared
2. Bluetooth
3. WI-FI

2.2 Multimedia

Multimedia dapat diartikan sebagai penggunaan beberapa media yang berbeda untuk menggabungkan dan menyampaikan informasi dalam bentuk text, audio, grafik, animasi, dan video. Sedangkan menurut wikipedia.org:

”*Multimedia is the use of several different media to convey information (text, audio, graphics, animation, video, and interactivity). Multimedia also refers to computer data storage devices, especially those used to store multimedia content*. “

Sistem dapat disebut Sistem Multimedia yaitu jika.

1. Kombinasi Media

   Sistem disebut sistem multimedia jika kedua jenis media (*continuous/ discrete*) dipakai. Contoh media diskrit : teks dan gambar, dan media kontinu adalah audio dan video.

2. *Independence*

   Aspek utama dari jenis media yang berbeda adalah keterkaitan antar media tersebut. Sistem disebut sistem multimedia jika tingkat ketergantungan/ keterkaitan antar media tersebut rendah.

3. *Computer-supported Integration*

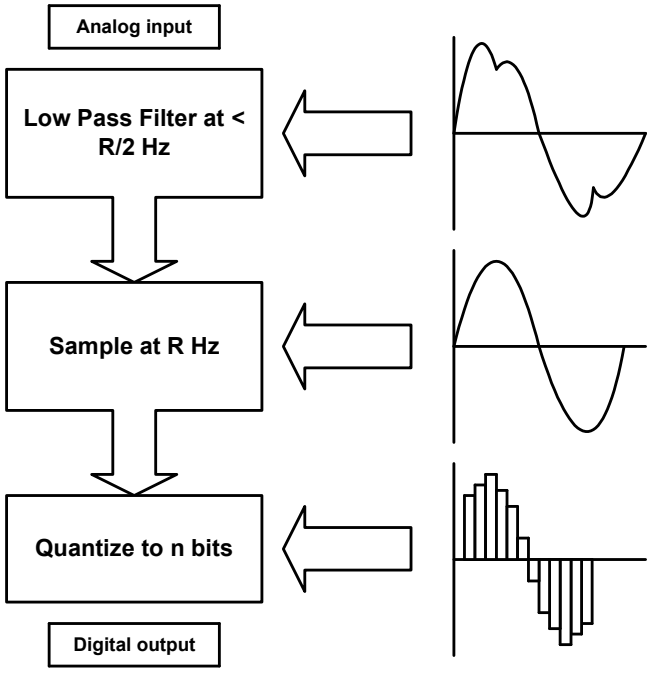
Sistem harus dapat melakukan pemrosesan yang dikontrol oleh komputer. Sistem dapat diprogram oleh sistem programmer/ user.

Karier Multimedia sangat dibutuhkan dalam berbagai bidang. Bidang-bidang tersebut antara lain, Pemasaran, animasi, desain grafis, elektronik publishing, editor, desain game, public relations, spesial efek, produksi video (video klip), dan web master.

2.3 Audio dan Video Digital

Suara yang didengar sehari-hari adalah gelombang analog. Gelombang ini berasal dari tekanan udara yang ada di sekelilingnya, yang dapat didengar dengan bantuan gendang telinga. Gendang telinga ini bergetar, dan getaran ini dikirim dan diterjemahkan menjadi informasi suara yang dikirimkan ke otak, sehingga dapat mendengar suara. Suara yang dihasilkan sewaktu berbicara berbentuk tekanan suara yang dihasilkan oleh pita suara. Pita suara ini akan bergetar, dan getaran ini menyebabkan perubahan tekanan udara, sehingga kita akan mengeluarkan suara.

Komputer hanya bisa mengenal bit “0” dan “1”. Kumpulan bit-bit ini kemudian diproses untuk menghasilkan informasi tertentu oleh komputer. Karena komputer hanya bisa memproses bit “0” dan “1” (digital) maka perlu adanya konversi dari sinyal analog menjadi sinyal digital yang dilakukan oleh *Analog-to-Converter* (ADC). Dan untuk memainkan file *audio* digital ini lagi, diperlukan konverter *Digital-to-audio* (DAC) agar suara dapat dikeluarkan melalui media elektronis analog seperti speaker. Biasanya proses ini berlangsung di *sound card*.

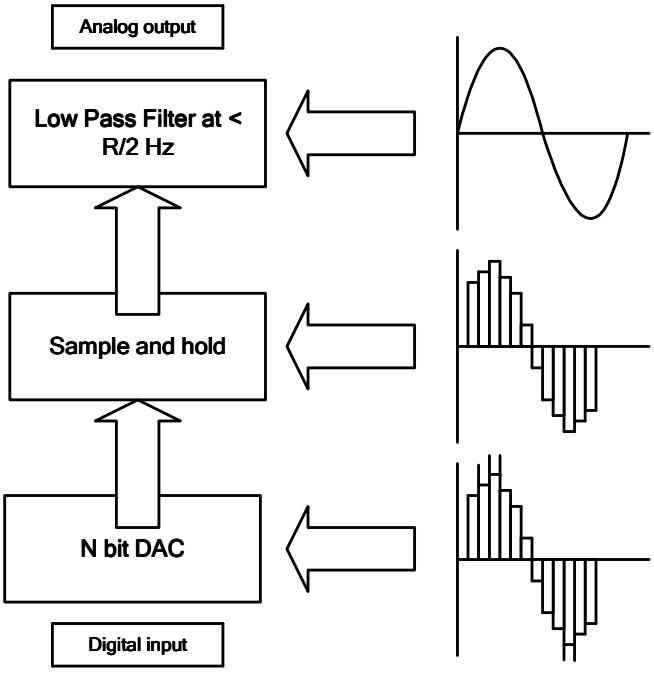


Gambar 2.1 Konversi sinyal analog ke digital

Pada Gambar 2.1 memperlihatkan proses pengubahan dari tegangan analog ke data digital terdiri atas beberapa tahap, yaitu:

a. Membatasi frekuensi sinyal yang akan diproses dengan *Low Pass Filter*.
b. Mencuplik sinyal analog ini (melakukan sampling) menjadi beberapa potongan waktu.

Cuplikan-cuplikan ini diberi nilai eksak, dan nilai ini diberikan dalam bentuk data digital.



Gambar 2.2 Konversi sinyal digital ke analog.

Pada Gambar 2.2 memperlihatkan proses sebaliknya pengubahan dari data digital menjadi tegangan analog terdiri atas beberapa tahap, yang ditunjukkan, yaitu:

a. Menghitung data digital menjadi amplitudo-amplitudo analog.
b. Menyambung amplitudo analog menjadi sinyal analog.
c. Memfilter keluaran dengan Low Pass Filter sehingga bentuk gelombang keluaran menjadi lebih mulus.

Proses pengubahan sinyal analog menjadi digital tadi harus memenuhi sebuah kriteria, yaitu kriteria Nyquist. Kriteria ini mengatakan bahwa untuk mencuplik sebuah sinyal yang memiliki frekuensi X Hertz, maka harus mencupliknya minimal dua kali lenih rapat, atau 2 Hertz. Jika tidak, sinyal tidak akan dapat dikembalikan ke dalam bentuk semula.

Kelebihan audio digital adalah kualitas reproduksi yang sempurna. Kualitas reproduksi yang sempurna yang dimaksud adalah kemampuannya untuk menggandakan sinyal audio secara berulang-ulang tanpa mengalami penurunan kualitas suara.

Dalam dunia audio digital, ada beberapa istilah diantaranya:

1. Jumlah kanal (*channel*)

   Yaitu jumlah kanal audio yang digunakan. Istilah penggunaan satu kanal audio disbut mono, sementara dua kanal yaitu kanal sebelah kiri dan kanan disebut stereo. Selain mono dan stereo ada juga jumlah kanal yang lebih banyak, yaitu *surround*. Jumlah kanal surround berkisar antara lima, tujuh bahkan lebih.

2. Laju pencuplikan (*sample rate*)

   Ketika *soundcard* mengubah audio menjadi data digital, *soundcard* akan memecah suara tadi menurut nilai menjadi ptongan-potongan sinyal dengan nilai tertentu. Proses sinyal ini bisa terjadi ribuan kali dalam satu satuan waktu. Banyaknya pemotongan dalam satu satuan waktu ini dinamakan laju pencuplikan (*sample rate*).

   Kerapatan laju pencuplikan ini menentukan kualitas sinyal analog yang akan dirubah menjadi data digital. Makin rapat laju pencuplikan ini, kualitas suara yang dihasilkan akan makin mendekati suara aslinya. Sebagai contoh, lagu yang tersimpan dalam Compact Disc Audio memiliki sample rate 44.1 kHz, yang berarti

lagu ini dicuplik sebanyak 44100 kali dalam satu detik untuk memastikan kualitas suara yang hampir sama persis dengan aslinya.

Tabel 2.2 Frekuensi sample dan kualitas suara yang dihasilkan

Sample rate (kHz)	Aplikasi
8	Telepon
11,025	Radio AM
16	Kompromi antara 11,025 dan 22,05 kHz
22,025	Mendekati radio FM
32,075	Lebih baik dari Radio FM
44,1	Audio Compact Disc (CD)
48	Digital Audio tape (DAT)

Perhatikan tabel 2.2 di atas. Tabel ini memiliki kaitan sample rate yang akan digunakan dengan kualitas suara yang dihasilkan. Rentang suara yang dapat didengar oleh manusia berkisar antara frekuensi 20 Hz sampai dengan 20000 Hz, sehingga besar rentang frekuensi adalah 19980 Hz. Hukum Nyquist berlaku, maka kita harus menggunakan frekuensi sampling minimal 2 kali rentang frekuensi tadi, yaitu 2 x 19980 Hz, sehingga frekuensi sampling minimal adalah 39960 Hz. Frekuensi sampling yang mencukupi adalah 44100 Hz.

3. Banyak bit dalam tiap cuplikan (*Bit per sample*)

Cuplikan-cuplikan yang telah disebutkan tadi diambil memiliki besaran amplitudo. Besaran amplitudo inilah yang akan disimpan dalam bit-bit digital. Banyaknya bit yang dapat dipakai untuk

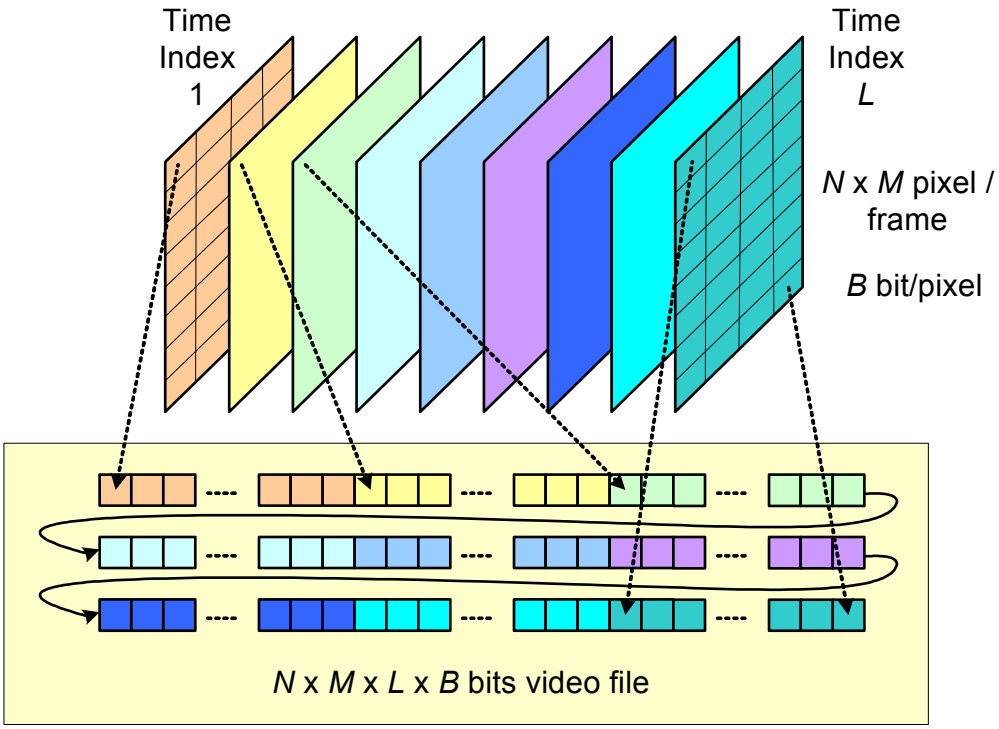
merepresentasikan besaran amplitudo ini dinamakan *bit per sample*. Makin banyak bit yang dipakai untuk merepresentasikan besaran amplitudo, makin halus besaran amplitudo yang akan dihasilkan. Sebagai contoh, suara 8-bit memiliki 2 pangkat 8 kemungkinan amplitudo, yaitu 256, dan suara 16-bit memiliki 65536 kemungkinan amplitudo.

4. Laju bit (*bit rate*)

   Laju bit atau *bit rate*(dengan satuan bit per detik) adalah perkalian antara jumlah kanal, frekuensi sampling (dengan satuan Hertz) dan bit per sample(dengan satuan bit). Dengan menggunakan informasi diatas, didapati bahwa bit rate yang dibutuhkan untuk menyimpan sebuah lagu stereo (dua kanal) dengan kualitas CD (frekuensi sampling 44100 Hz, 16-bit) adalah 2 x 44100 Hz x 16 = 1411200 bit per detik.

Suara (Sound) fenomena fisik yang dihasilkan oleh getaran benda getaran suatu benda yang berupa sinyal analog dengan amplitudo yang berubah secara kontinyu terhadap waktu. Range pendengaran manusia yaitu berada diantara 20 Hz - 20 KHz. Suara yang berada pada range pendengaran manusia disebut Audio.

Sinyal video digital adalah kumpulan citra digital (disebut *frame*) yang di 'tayangkan' secara berturut menurut indeks waktu. Adapun sinyal video digital yang dijelaskan pada gambar 2.3.



Gambar 2.3 Sinyal Video Digital

2.4 Kompresi Data

Proses kompresi telah lama dikenal dalam dunia komputer. Hal ini terlihat dengan banyaknya teknik kompresi *audio* yang muncul hingga sekarang ini. Proses kompresi umumnya bertujuan untuk menempatkan ukuran data sehingga dapat menghemat ruang penyimpanan pada media penyimpanan data seperti *harddisk*. Proses kompresi juga berdampak pada kecepatan pemrosesan data. Hal ini disebabkan proses dekompresi yang harus dilakukan terlebih dahulu sebelum memperoleh data yang asli.

Kompresi berarti memampatkan/mengecilkan ukuran. Kompresi data adalah proses mengkodekan informasi menggunakan bit atau *information-bearing unit* yang lain yang lebih rendah daripada representasi data yang tidak terkodekan dengan suatu sistem *enkoding* tertentu. Contoh kompresi sederhana yang biasa kita lakukan misalnya adalah menyingkat kata-kata yang sering digunakan tapi sudah memiliki konvensi umum. Misalnya: kata "yang" dikompres menjadi kata "yg".

Secara umum teknik kompresi dibagi menjadi tiga macam yaitu *lossless*, *lossy*, dan *hybrid*. *Lossless* merupakan golongan teknik kompresi dimana data keluaran proses dikompresi sama dengan data masukan proses kompresi. *Lossy* merupakan golongan teknik kompresi dimana data keluaran pada proses dekompresi berbeda dengan data masukan proses kompresi. *Hybrid* merupakan golongan teknik kompresi yang dapat mengembalikan data secara sempurna ataupun berbeda pada proses dekompresi.

Proses kompresi pada data *audio* pada umumnya menggunakan teknik kompresi *lossy* seperti mp3. Namun, dengan menggunakan teknik kompresi *lossy*, kualitas data *audio* tersebut menjadi berkurang. Sehingga dikembangkan metode kompresi *audio* dengan menggunakan teknik *lossless.* MPEG sendiri hingga saat ini masih mengembangkan metode kompresi *lossless* pada *audio.*

2.5 Kompresi Audio

Kompresi audio adalah salah satu bentuk kompresi data yang bertujuan untuk mengecilkan ukuran file audio dengan metode *Lossy* dan *Lossless.* Terdapat beberapa kendala kompresi audio, yaitu :

1. Perkembangan sound recording yang cepat dan beranekaragam.
2. Nilai dari audio sample berubah dengan cepat.

Losless audio codec tidak mempunyai masalah dalam kualitas suara, penggunaannya dapat difokuskan pada :

1. Kecepatan kompresi dan dekompresi.
2. Derajat kompresi.
3. Dukungan hardware dan software.

Lossy audio codec penggunaannya difokuskan pada:

1. Kualitas audio.
2. Faktor kompresi.
3. Kecepatan kompresi dan dekompresi.
4. *Inherent latency of algorithm* (penting bagi real-time streaming).
5. Dukungan hardware dan software.

2.6 Kompresi Video

Kompresi video adalah salah satu bentuk kompresi data yang bertujuan untuk mengecilkan ukuran file video dengan metode *Lossy* dan *Lossless*. Video memiliki 3 dimensi yaitu 2 dimensi spatial (horizontal dan vertical) dan 1 dimensi waktu. Didalam video terdapat 2 hal yang dapat dikompresi yaitu frame (stillimage) dan audionya.

Data video memiliki *redudancy spatial* (warna dalam still image) yaitu penghilangan yang dilakukan dengan mengambil keuntungan dari fakta bahwa mata manusia tidak terlalu dapat membedakan warna dibandingkan dengan *brightness* sehingga image video dapat dikompresi dan *redundancy temporal*

(perubahan antar frame) adalah penghilangan yang dilakukan dengan mengirimkan dan mengkode frame yang berubah saja sedangkan data yang sama masih disimpan.

2.7 Kompresi Image

Kompresi image atau citra adalah aplikasi kompresi data yang dilakukan terhadap citra digital dengan tujuan untuk mengurangi redundansi dari data-data yang terdapat dalam citra sehingga dapat disimpan atau ditransmisikan secara efisien.

Kompresi pada citra tetap sama dibagi menjadi 2 yaitu :

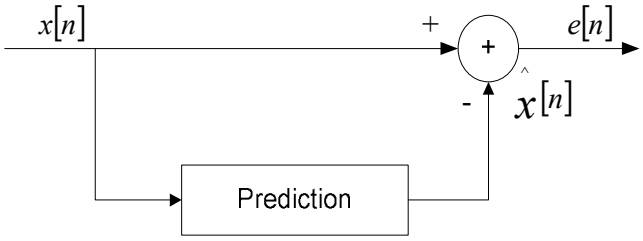
1. *lossy* dimana Ukuran file citra menjadi lebih kecil dengan menghilangkan beberapa informasi dalam citra asli. Teknik ini mengubah detail dan warna pada file citra menjadi lebih sederhana tanpa terlihat perbedaan yang mencolok dalam pandangan manusia, sehingga ukurannya menjadi lebih kecil. Biasanya digunakan pada citra foto atau image lain yang tidak terlalu memerlukan detail citra, dimana kehilangan bit rate foto tidak berpengaruh pada citra.
2. *lossless* dimana teknik ini kompresi yang dilakukan pada citra tersebut tidak ada data yang hilang satupun sehingga menghasilkan data citra yang baik dan sama pada saat proses dekompresi, biasanya teknik ini digunakan pada citra medis.

2.8 Framing

File video, audio, citra, text yang dikompres terlebih dahulu disimpan dalam *buffer* memori. Dalam *buffer*, file dibagi menjadi frame-frame data tersebut. Penentuan ukuran frame yang digunakan sesuai dengan ukuran frame yang disarankan oleh MPEG. Dalam setiap frame terdapat sejumlah sampel data yang tergantung pada ukuran frame. Hasil dari framing inilah yang menjadi inputan bagi proses *Polynomial Linear Prediction Coding* (LPC).

2.8.1 *Linear Prediction*

Prinsip dasar dari *Linier Prediction* adalah bahwa sebuah sample dari sinyal $x[n]$ dapat diperkirakan sebagai kombinasi linier dari sample sebelumnya $x[n-1],\ x[2]$ dst.



Gambar 2.4 *Linier Prediction System*

Pada Gambar 2.4 memperlihatkan prinsip dasar dari *Linear Prediction.* Dalam proses sinyal diskrit, sebuah filter digital *H(z)* diasumsikan memiliki p pole dan *q* zero, yang berarti dengan inputan $x[n]$, prediksi sinyal tersebut $\hat{x}[n]$ dapat dimodelkan dengan kombinasi *q* sampel input sebelumnya dan *p+1* sampel output sebelumnya dalam sebuah sistem filter distkrit :

$$\hat{x}[n] = \sum_{k=0}^{p} b_k x[n-k] + \sum_{k-1}^{q} a_k \hat{x}[n-k] \tag{2.1}$$

yang ekivalen dengan :

$$H(z) = \frac{\hat{X}(z)}{X(z)} = \frac{\sum_{k=0}^{p} b_k z^{-k}}{1 - \sum_{k=1}^{q} a_k z^{-k}} \tag{2.2}$$

Persamaan di atas merupakan bentuk umum dari Filter *Infinite Respons Impulse (IIR)* orde ke-*p*. Model seperti itu dengan memiliki pole dan zero disebut juga model *ARMA (Autogressive Moving Average)*. Jika kita membuat koefesien $a_{1,} a_2, ... a_3$ sama dengan nol, maka persamaan di atas menjadi sebuah sistem *Finite Impuls Response* (FIR).

$$\hat{x}[n] = b_0 x[n] + b_1 x[n-1] + ... b_p x[n-p] = \sum_{k=0}^{p} b_k x[n-k] \tag{2.3}$$

2.8.2 FIR *Linier Prediction*

Umumnya, sistem *linier prediction* pada literatur diasumsikan menggunakan FIR karena kemampuannya dalam memprediksi secara akurat dan kecepatan dalam proses komputasi. Bentuk umum dari fungsi sistem FIR Linier Prediction orde ke-*p* merupakan polynomial.

$$P(z) = \sum_{k=1}^{p} b_k z^{-k} \tag{2.4}$$

dimana setiap zero pada *z = 0* diabadikan karena tidak berpengaruh pada spektral magnitude dan hanya menambah phasa linier saja.

2.8.3 *Error Prediction*

Output dari Linier Prediction merupakan selisih antara sinyal input original $x[n]$ dan prediksinya $\hat{x}[n]$,

$$e[n] = x[n] - \hat{x}[n] = x[n] - \sum_{k=1}^{p} b_k x[n-k] \tag{2.5}$$

dimana sinyal selisih $e[n]$ disebut sebagai *residual* atau eror prediksi. Dari persamaan di atas terlihat bahwa urutan dari eror prediksi merupakan output dari sistem yang memiliki fungsi transfer :

$$A(z) = 1 - \sum_{k=1}^{p} b_k z^{-k} \tag{2.6}$$

yang merupakan inverse filter $H(z) = Az^{-1}$. Pemasangan dasar dari linier Prediction yaitu bagaimana menentukan koefesien $\{b_k\}$ untuk mendapatkan prediksi yang optimal dari sinyal input.

Koefesien predksi yang optimal dipilih sedemikian rupa sehingga energi (rata-rata) dari sinyal residual dapat seminimal mungkin yang didefinisikan sebagai :

$$E = \sum_{n=-\infty}^{\infty} e^2[n] \tag{2.7a}$$

$$= \sum_{n=-\infty}^{\infty} (x[n] - \hat{x}[n])^2 \tag{2.7b}$$

$$= \sum_{n=-\infty}^{\infty} (x[n] - \sum_{k=1}^{p} b_k x[n-k]\ ]^2 \tag{2.7c}$$

Dengan membuat $\partial E / \partial b_k = 0 \; k = 1,2,3..., p,$ sebagai kondisi yang memungkinkan untuk memperoleh energi minimum, diperoleh persamaan linier *p*.

$$\sum_{n=-\infty}^{\infty} (x[n-1]x[n] = \sum_{k=1}^{p} b_k \sum_{n=-\infty}^{\infty} x[n-i]x[n-k], \; i = 1,2,3..., p \tag{2.8}$$

Jika kita mendefinisikan

$$\phi(i,k) = \sum_{n=-\infty}^{\infty} x[n-i]x[n-k] \tag{2.9}$$

Maka persamaan (2.8) dapat ditulis dengan

$$\sum_{k=1}^{p} b_k \phi(i,k) = \phi(i,0). \qquad i = 1,2,3..., p \tag{2.10}$$

Dari persamaan (2.7), (2.8), dan (2.9) *minimum-squared* dari error prediksi dapat ditunjukkan dengan

$$E = \sum_{-\infty}^{\infty} x^2[n] - \sum_{k=1}^{p} b_k \sum_{-\infty}^{\infty} x[n] - x[n-k] \tag{2.11}$$

$$= \phi(0,0) - \sum_{k=1}^{p} b_k \phi(0,k) \tag{2.12}$$

2.8.4 Metode Korelasi

Untuk menyelesaikan persamaan (2.10), limit dari penjumlahan pada persamaan (2.7) dan (2.8) harus beradapada interval yang berhingga. Salah satu pendekatan dasar dalam menentukan limit tersebut yaitu metode autokorelasi. Pendekatan ini mengasumsikan bahwa sinyal $x[n]$ memiliki durasi yang terbatas dengan panjang *N* sehingga $x[n]$ = 0 pada $0 \leq n \leq N - 1$. dengan demikian, sinyal residual yang diperoleh melalui filter Linear

Prediction orde ke-*p* tidak sama dengan nol pada interval $0 \leq n \leq N - 1 + p$, sehingga energi residual dan persamaan (2.9)dapat ditulis sebagai

$$E = \sum_{n=0}^{N+p-1} e^2[n], \tag{2.12a}$$

$$\phi(i,k) = \sum^{N+p-1} x[n-1]x[n-k] \tag{2.12b}$$

Dimana $i = 1, 2, \ldots, p$ dan $k = 0,1, \ldots, p$ Karena $\varphi(i,k)$ dari persamaan (1.4.12b) identik dengan fungsi autokorelasi untuk $(i - k)$, persamaan linier (3.2.10) dapat ditulis sebagai persamaan biasa yang disebut persamaan *Wiener-Hopf,*

$$\sum_{k=1}^{p} a_k R(|i-k|) = R(i), \qquad i = 1,2,\ldots, p, \tag{2.13}$$

Dimana dalam bentuk materik adalah sebagai berikut :

$$\begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ . \\ . \\ . \\ R(p) \end{bmatrix} = \begin{bmatrix} R(0) & R(1) & \ldots & R(p-1) \\ R(1) & R(0) & \ldots & R(p-2) \\ R(2) & R(1) & \ldots & R(p-3) \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ R(p-1) & R(p-2) & \ldots & (R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ . \\ . \\ . \\ a_p \end{bmatrix} \tag{2.14}$$

Sama dengan persamaan (2.11), minimum mean-squared dari error prediksi menjadi

$$E = R(0) - \sum_{k=1}^{p} a_k R(k), \tag{2.15}$$

Dimana *R(k)* merupakan korelasi ke-*k* sinyal asal dan *R(0)* merupakan variasi $\left(\sigma_x^{\,2}\right)$ dari sinyal asal. Pada persamaan (2.14), materiks autokorelasi *p x*

p merupakan sebuah matriks *Toeplitz* yang mana simetris dan semua elemen pada diagonal sama. Kondisi ini dimanfaatkan untuk menyelesaikan persamaan (2.13).

2.8.5 Metode *Levinson-Durbin*

Persamaan Wiener-Hopf dapat diselesaikan secara efisien dengan menggunakan metode Levinson-Durbin, yang mana dalam operasi berikut dilakukan secara rekursif untuk $m = 1, 2, ..., p$:

$$k_m = \frac{\left[R(m) - \sum_{k=1}^{m-1} a_{m-1}[k] R(m-k) \right]}{E_{m-1}} \tag{2.16a}$$

$$a_m[m] = k_m, \tag{2.16b}$$

$$a_m[k] = a_{m-1}[k] - k_m a_{m-1}[m-k], \tag{2.16c}$$

$$E_m = (1 - k_m^2) E_m - 1 \tag{2.16d}$$

Dimana kondisi awal $E_0 = R(0)$ dan $a_0 = 0$. Koefisien refleksi k_m menjamin filter linier Prediction $H(z)$ stabil sehingga untuk setiap putaran *m* koefisien $a_m[k]$ menggambarkan predictor linier orde ke-*m* yang optimal dan error minimum E_m berkurang dengan faktor $(1 - k_m^2)$. Oleh karena itu, kondisi yang perlu untuk sistem stabil yaitu $|k_m| \leq 1$.

2.8.6 Polinomial LPC

Berdasarkan penjelasan di atas, terlihat bahwa untuk memperoleh koefisien predikat yang optimal membutuhkan perhitungan yang rumit. Untuk menghindari perhitungan yang rumit dalam mendapatkan koefisien prediksi

yang optimal dalam Linier Prediction, dilakukan pendekatan dengan menggunakan metode prediksi FIR yang sederhana dengan hanya menggunakan koefisien integer saja. Pendekatan ini pertama kali diajukan pada *Shorten* [15].

Tujuan dari *polinomial prediction* yaitu untuk memodelkan suatu koefisien FIR *h(k), k* = 1, 2, ..., *p*, dimana *p* merupakan orde FIR, yang memprediksi polinomial sinyal input dengan tepat.

Dengan menggunakan polinomial LPC, sinyal prediksi diperoleh melalui persamaan-persamaan dibawah ini :

$$\hat{x}_0[n] = 0 \tag{2.17a}$$

$$\hat{x}_1[n] = x[n-1] \tag{2.17b}$$

$$\hat{x}_2[n] = 2x[n-1]) - x[n-2] \tag{2.17c}$$

$$\hat{x}_3[n] = 3x[n-1]) - 3x[n-2] + x[n-3] \tag{2.17d}$$

Melalui persamaan di atas, sinyal residual, $e[n] = x[n] - \hat{x}[n]$, dapat diperoleh tanpa perhitungan yang rumit

$$e_0[n] = [n] = x[n] \tag{2.18a}$$

$$\begin{aligned} e_1[n] &= [n] = e_0[n] - e_0[n-1] \\ &= x[n]x[n-1] \end{aligned} \tag{2.18b}$$

$$\begin{aligned} e_2[n] &= e_1[n] - e_1[n-1] \\ &= (x[n] - x[n-1]) - (x[n-1] - x[n-2]) \\ &= x[n] - (2x[n-1]) - x[n-2]) \end{aligned} \tag{2.18c}$$

$$e_3[n] = e_2[n] - e_2[n-1]$$

$$= (x[n] - 2x[n-1]) - + x[n-2]) - (x[n-1] - 2x[n-2]) + x[n-3])$$

$$= x[n] - (3x[n-1]) - 3x[n-2] + x[n-3]) \quad (2.18d)$$

Untuk setiap frame, keempat error pediksi $e_0[n], e_1[n], e_2[n],$ dan $e_3[n]$ dihitung untuk setiap sampel yang ada pada frame tersebut. Kemudian nilai mutlak dari error prediksi ini dijumlahkan untuk setiap framena. Error prediksi dengan jumlah terkecil ditetapkan sebagai prediksi yang paling bagus dan order prediksi digunakan pada frame tersebut. Begitu juga proses untuk frame yang lainnya. Dengan demikian, setiap frame dapat memiliki order prediksi yang berbeda debfab frame lainnya. Error prediksi ini menjadi masukan bagi proses *Rice Coding.*

2.8.7 *Rice Coding*

Rice Coding merupakan suatu bentuk *entropy coding,* yang merupakan bentuk lain dari *Huffman Coding* untuk menyelesaikan kasus khusus. Apabila inputan memiliki pola distribusi data seperti distribusi *Laplacian* dan distribusi *Geometri* maka *Huffman coding* memberikan hasil yang kurang optimal dalam melakukan kompresi data inputan tersebut. Sebaliknya *Rice Coding* memberi hasil yang lebih baik dibandingkan *Huffman*.

2.8.8 Parameter Rice Coding

Dalam *rice coding* terdapat parameter *k,* yang digunakan untuk menginisialisasi jumlah bit yang dikompresi. Jika nikai *k* yang diinisialisasi

terlalu besar ataupun terlalu kecil maka menghasilkan output kompresi yang kurang optimal. Oleh karena itu nilai *k* sangat berpengaruh untuk melakukan kompresi. Nilai *k* yang optimum biasanya diperoleh dari perhitungan *k*=[*log(n)/log(2)*] atau dari kemunculan setiap data.

Contoh : 10, 14, 15, 46 dalam biner menjadi 1010, 1110, 1111, 10110. dari data inputan ditemukan bahwa jumlah bit yang paling sering muncul adalah 4 (3 bilangan pertama). Maka dipilih $k = 4$.

Langkah-langkah komposisi dalam *rice coding* untuk data inputan *n* dan *k* bit yang di-*encode,* adalah :

1. Bit tanda, bit 1 menandakan bahwa *n* adalah positif dan bit 0 menandakan bahwa *n* adalah negatif.
2. Bit 0, misalkan $x = n/(2^k)$ tambahkan bit 0 sebanyak *x*.
3. Akhiri dengan bit 1
4. *k Lsb*, ambil lsb sejumlah *k* dari inputan *n*
5. Satukan semua bit, [bit tanda], [bit nol][1][bit lsb].

Proses dekompresi sangat sederhana dan sangat cepat daripada proses kompresi. Secara algoritma dapat dijelaskan sebagai berikut :

1. $x \leftarrow 1$ /* periksa jumlah bit yang ada*/
2. $n \leftarrow 1$ /* bit sekarang */
3. $K[1][0] = K[1][1] \leftarrow 1$ /* model inisialisasi */
4. - Jika kode lebih besar, maka:

   $K[n][e] \leftarrow x$; $K[x][1] \leftarrow K[n][1]$

- Jika kode lebih kecil, maka:

$K[n][e] \leftarrow x$; $K[x][0] \leftarrow K[n][0]$

5. lakukan proses pembaharuan nilai bit:

   a. $n \leftarrow x$

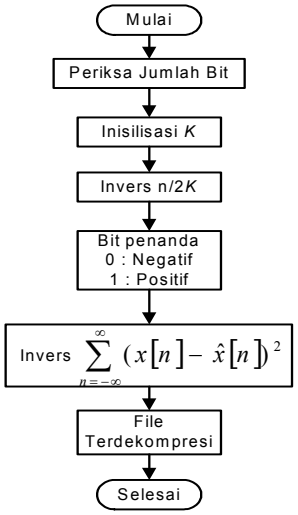
   b. $n \leftarrow K[n][e]$

Jika frekuensi transisi dari suatu bit *n* ke bit sebelumnya, yaitu bit *x*, sangat tinggi, maka bit *n* dapat didekompresi. Bit yang didekompresi diberi simbol *n*'. Aturan dekompresi adalah sebagai berikut :

1. Semua transisi dari bit *n* dikirim ke bit *n*'. Semua transisi dari bit lain ke bit *n* tidak berubah.
2. Jumlah transisi yang keluar dari *n*' harus mempunyai rasio yang sama (antara 0 dan 1) dengan jumlah transisi yang keluar dari *n*.

Jumlah transisi yang keluar dari *n* dan *n*' diatur supaya mempunyai nilai yang sama dengan jumlah transisi yang masuk

Berikut merupakan diagram alir dari proses dekompresi :



Gambar 2.5 Diagram Alir proses Dekompresi

Gambar 2.5 menjelaskan proses dekompresi, pertama-tama periksa jumlah bit yang akan didekompresi, kemudian inisialisasikan dengan *K* sama seperti proses kompresi, lalu hitung *invers n/2K*, kemudian beri penanda bit yaitu 0 menandakan negatif dan 1 untuk positif, lalu hitung

$$Invers \sum_{n=-\infty}^{\infty} (x[n] - \hat{x}[n])^2$$

Dengan menggunakan persamaan berikut :

$$E = \sum_{n=-\infty}^{\infty} e^2[n] \tag{2.7a}$$

$$= \sum_{n=-\infty}^{\infty} (x[n] - \hat{x}[n])^2 \tag{2.7b}$$

$$= \sum_{n=-\infty}^{\infty} (x[n] - \sum_{k=1}^{p} b_k x[n-k]\ ]^2 \tag{2.7c}$$

2.9 Jaringan Komputer

Secara umum jaringan komputer adalah beberapa komputer yang saling berhubungan dan melakukan komunikasi satu dengan yang lain menggunakan perangkat keras jaringan (Ethernet card, token ring, bridge, modem dan sebagainya).

Komputer yang berada dalam suatu jaringan dapat melakukan tukar-menukar informasi atau data dengan komputer lain yang berada dalam jaringan tersebut. Pengguna suatu komputer dapat melihat dan mengakses data pada komputer lain dalam jaringan apabila dilakukan *file sharing*.

Pada era 70-an, jaringan komputer biasanya hanya terdapat pada perusahaan-perusahaan besar. Jaringan tersebut menghubungkan kantor-kantor dan departemen-departemen dalam lingkungan perusahaan tersebut (intranet)

Pada tahun 1974, internet muncul dan membawa perubahan besar. Proyek ARPA semula hanya disiapkan untuk membangun jalur komunikasi antara beberapa universitas, pangakalan-pangkalan militer dan perusahaan yang terlibat dalam proyek tersebut.

Saat ini, jaringan internet sudah digunakan orang di berbagai belahan bumi. Sayangnya, selain membawa dampak positif, internet juga membawa dampak negatif. Oleh karena itu, pengguna harus selektif memanfaatkan informasi dan fasilitas yang ada. Perkembangan jaringan komputer di Indonesia sendiri telah mengalami kemajuan pesat meskipun menemui beberapa kendala di dalamnya.

2.9.1 Tipe-Tipe Jaringan Komputer

Jaringan komputer dapat dibedakan berdasarkan cakupan geografisnya, yaitu sebagai berikut:

1. LAN (*Local Area Nerwork*)

Jaringan LAN digunakan untuk menghubungkan komputer yang berada di dalam suatu area yang kecil, misalnya di dalam suatu gedung perkantoran atau kampus. Jarak antar komputer yang dihubungkannya bisa mencapai 5 sampai 10 km. Suatu LAN biasanya bekerja pada

kecepatan mulai 10 Mbps sampai 100 Mbps. LAN menjadi populer karena memungkinkan banyak pengguna untuk memakai sumber daya secara bersama-sama. Contoh dari sumber daya yang dapat digunakan itu misalnya suatu *mainframe, file server*, printer, dan sebagainya.

2. MAN (*Metropolitan Area Network*)

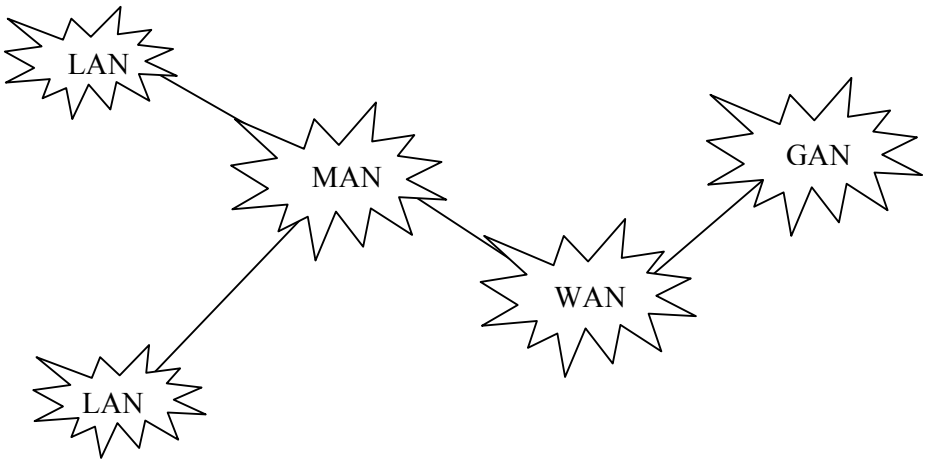
MAN merupakan suatu jaringan yang cakupannya meliputi suatu kota. MAN menghubungkan LAN-LAN yang lokasinya berjauhan. Jangkauan MAN bisa mencapai 10 km sampai beberapa ratus kilometer. suatu MAN biasanya bekerja pada kecepatan 1,5 sampai 150 Mbps.

3. WAN (*Wide Area Network*)

WAN dirancang untuk menghubungkan komputer-komputer yang terletak pada suatu cakupan geografis yang luas, seperti hubungan dari suatu kota ke kota lain di dalam suatu negara. Cakupan WAN bisa meliputi 100 km sampai 1.000 km, dan kecepatan antar kota bisa bervariasi antara 1,5 Mbps sampai 2,4 Gbps. Dalam WAN, biaya untuk peralatan transmisi sangat tinggi, dan biasanya jaringan WAN dimiliki dan dioperasikan sebagai suatu jaringan publik.

4. GAN (*Global Area Network*)

GAN merupakan suatu jaringan yang menghubungkan negara-negara di seluruh dunia. Kecepatan GAN bervariasi mulai dari 1,5 Mbps sampai dengan 100 Gbps dan cakupannya mencapai ribuan kilometer. Contoh yang sangat baik dari GAN adalah Internet.



Gambar 2.6 Interaksi antara LAN, MAN, WAN, dan GAN

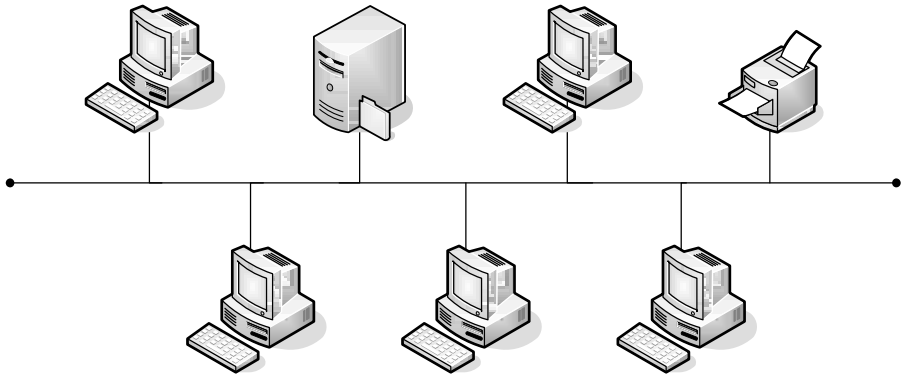
Pada Gambar 2.6 memperlihatkan interaksi antara jaringan LAN, MAN, WAN dan GAN.

2.9.2 Topologi Jaringan Komputer

Berikut ini akan dijelaskan mengenai topologi fisik yang digunakan di dalam jaringan lokal diantaranya :

1. *Linear Bus* (Garis Lurus)

Topologi *linear Bus* (Garis lurus) terdiri dari satu jalur kabel utama dimana masing-masing ujungnya diberikan sebuah *terminator*. Semua *nodes* pada jaringan (*file server, workstation,* dan perangkat lainnya) terkoneksi sebuah kabel utama (*backbone*). Jaringan-jaringan *Ethernet* dan *Local talk* menggunakan topologi *linear bus* ini.



Gambar 2.7 Topologi *Linear Bus* (Garis Lurus)

Pada Gambar 2.7 Memperlihatkan jaringan toplologi *Linear Bus.*

Adapun kelebihan dari topologi *Linear Bus* (Garis Lurus) adalah :

1. Mudah dalam konfigurasi komputer atau perangakat lain ke dalam sebuah kabel utama.
2. Tidak terlalu banyak menggunakakn kabel dibandingkan dengan topologi star atau bintang.

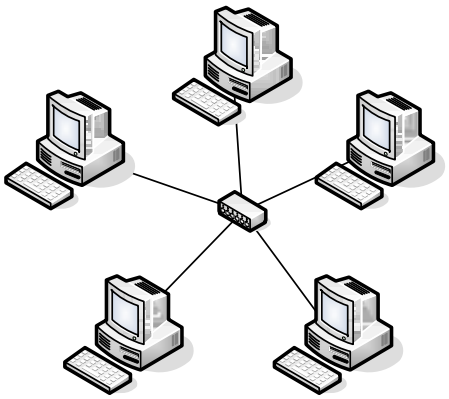
Kekurangan dari topologi *Linear Bus* (Garis Lurus) adalah:

1. Seluruh jaringan akan mati jika ada kerusakan pada kabel utama (backbone).
2. Membutuhkan terminator pada kedua sisi dari kabel utamanya.
3. Sangat sulit dalam mengidentifikasi permasalahan jika jaringan sedang jatuh atau rusak.
4. Sangat tidak disarankan dipakai sebagai salah satu solusi pada penggunaan jaringan di gedung besar.

2. Star (Bintang)

Topologi model ini dirancang yang mana setiap *nodes* (*file server, workstation*, dan perangkat lainnya) terkoneksi ke jaringan melewati sebuah *concentrator*.

Data yang dikirim ke jaringan lokal akan melewati *concentrator* sebelum melanjutkan ke tempat tujuannya. Concentrator akan mengatur dan mengendalikan keseluruhan fungsi jaringan, dan juga bertindak sebagian *repeater* (penguat aliran data). Konfigurasi pada jaringan model ini menggunakan kabel *Twiated Pair*, dan dapat digunakan pula kabel *coaxial* atau kabel *fibre optic*.



Gambar 2. 8 Topologi *Star* (Bintang)

Pada Gambar 2.8 Memperlihatkan jaringan toplologi *Star* (Bintang). Adapun kelebihan dari Topologi *Star* (Bintang) adalah :

1. Mudah dalam pemasangan dan pengkabelan.
2. Tidak akan mengakibatkan gangguan pada jaringan pada jaringan ketika akan memasang atau memindahkan perangkat jaringan lainnya.
3. Mudah untuk mendeteksi kesalahan dan memindahkan perangkat-perangkat lainnya.

Kekurangan dari topologi *star* (Bintang)

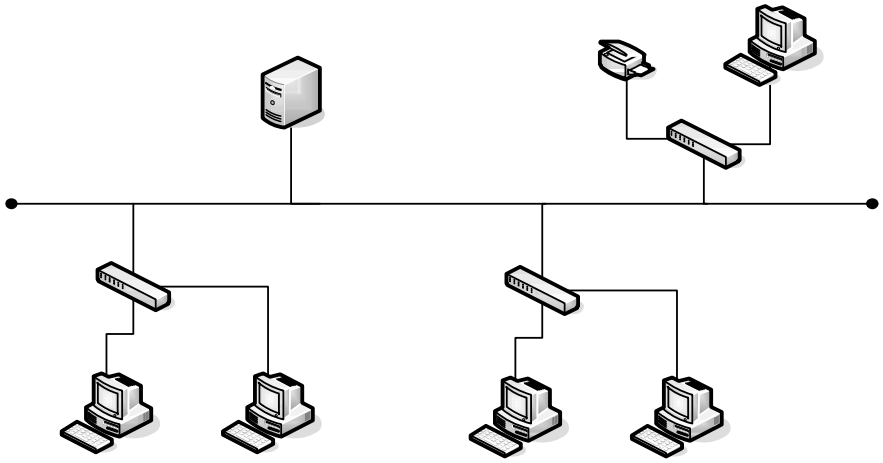
1. Membutuhkan labih banyak kabel dari pada *linear bus*.
2. Membutuhkan *concentrator*, dan bilamana *consentrator* tersebut rusak maka semua node yang terkoneksi tidak dapat terdeteksi.
3. Lebih mahal daripada topologi *linear bus*, karena biaya untuk pengadaan *concentrator*.

3. *Ring* (Cincin)

Topologi *Ring* (Cincin) menggunakan teknik konfigurasi yang sama denga topologi *star* tetapi pada pada topologi ini terlihat bahwa jalur media transmisi menyerupai suatu lingkaran tertutup menyerupai cincin (Lingkaran), sehingga diberi nama topologi bintang dalam lingkaran atau *star-wired ring*.

1. *Tree (Pohon)*

Topologi model ini merupakan perpaduan antara topologi *Linear Bus* dan *Star*, yang mana terdiri dari kelompok-kelompok dari *workstation* dengan konfigurasi *star* yang terkoneksi ke kabel utama yang menggunakan topologi *Linear Bus*. Topologi ini memungkinkan untuk pengembangan jaringan yang telah ada, dan memungkinkan untuk mengkonfigurasi jaringan sesuai dengan kebutuhan.



Gambar 2. 9 Topologi *Tree* (Pohon)

Pada Gambar 2.9 Memperlihatkan jaringan Topologi *Tree* (Pohon). Adapun kelebihan dari Topologi *Tree* (Pohon) adalah :

1. Proses konfigurasi jaringan dilakukan dari titik ke titik pada masing-masing segmen.
2. Didukung oleh banyak perangkat keras dan perangkat lunak

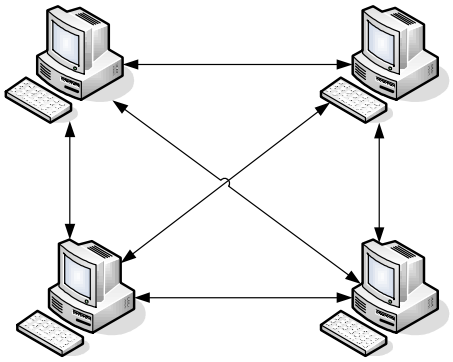
Kekurangan dari topologi *Tree* (Pohon)

3. Keseluruhan panjang kabel pada tiap-tiap segmen dibatasi oleh tipe kabel yang digunakan.
4. Jika jaringan utama (backbone) rusak, maka keseluruhan segmen ikut rusak juga.
5. Sangat relative sulit untuk di konfigurasi dan proses pengkabelannya dibandingkan topologi jaringan lain.

2.9.3 Model Hubungan pada LAN

1. *Peer-to-Peer*

   Model hubungan *peer to peer* memungkinkan user membagi sumber daya yang ada di komputer baik itu berupa *file*, layanan printer dan lain-lain serta mengakses sumber daya yang terdapat pada komputer lain. Namun model ini tidak mempunyai sebuah *file server* atau sumber daya yang terpusat. Didalam model hubungan *peer to peer* ini, seluruh komputer adalah sama, yang mana mempuyai kemampuan yang sama untuk memakai sumber daya yang tersedia di dalam jaringan model ini di desain untuk jaringan berskala kecil dan menengah.



Gambar 2. 10 Model Hubungan *Peer to Peer*

Pada Gambar 2.10 Memperlihatkan model jaringan *Peer to Peer*. Adapun kelebihan dari Model Hubungan *Peer to Peer* adalah :

1. Tidak terlalu mahal, karena tidak membutuhkan *dedicated file server*.
2. Mudah dalam konfigurasi programnya, hanya tinggal mengatur untuk operasi model hubungan *peer to peer*.

Kekurangan dari Hubungan Peer to peer

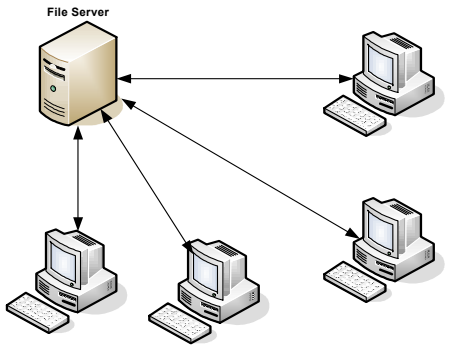
1. Tidak terpusat, terutama untuk penyimpanan data dan aplikasi
2. Tidak aman, karena tidak menyediakan fasilitas untuk keperluan itu.

2. Client/Server

Model hubungan *Client/Server* memungkinkan jaringan untuk mensentralisasi fungsi dan aplikasi kepada satu atau dua *dedicated file server*. Sebuah *file server* menjadi jantung dari keseluruhan sistem, memungkinkan untuk

mengakses sumber daya dan menyediakan keamanan. *Workstation* yang berdiri sendiri dapat mengambil sumber daya yang ada pada *file server*. Model hubungan

ini, menyediakan mekanisme untuk mengintegrasikan seluruh komponen yang ada di jaringan dan memungkinkan banyak penggunaan secara bersama-sama memakai sumber daya pada *file server*.



Gambar 2. 11 Model Hubungan *Client/server*

Pada Gambar 2.11 Memperlihatkan model jaringan *Client/Server*.

Adapun kelebihan dari Model Hubungan *Client/Server* adalah :

1. Terpusat (sumber daya dan keamanan data dikontrol melalui server)
2. Skalabilitas
3. Fleksibel
4. Teknologi baru dengan mudah terintegrasi ke dalam system
5. Keseluruhan komponen (*Client, network, server*) dapat bekerja bersama.

Kekurangan model hubungan *client/server*:

1. Mahal
2. Membutuhkan investasi untuk dedicated *file server*
3. Perbaikan (Jaringan besar membutuhkan seorang staff untuk mengatur agar sistem berjalan secara efisien)
4. Berketergantugan.
5. Ketika server jatuh mengakibatkan keseluruhan operasi pada *network* akan jatuh pula.

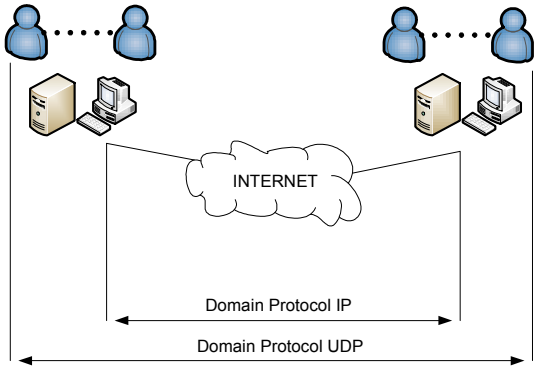
2.10 Protokol

Agar komputer-komputer dalam sebuah jaringan dapat berkomunikasi maka dibutuhkan sebuah protokol atau suatu aturan standar komunikasi baik antar komputer maupun antara jaringan komputer. *TCP*/*IP* (*Transmission Control Protocol* / *Internet Protocol*) ini menjadi standar protokol yang digunakan pada jaringan internet, karena *TCP*/*IP* dikembangkan untuk dapat diterapkan dihampir segala jenis *platform* komputer, biasa dikenal dengan konsep *open system*.

Protokol *TCP*/*IP* terdapat 2 protokol pada lapisan port yaitu :

2.10.1 UDP (*User Datagram Protocol*)

Bentuk proses bisa saja membentuk *client-server*



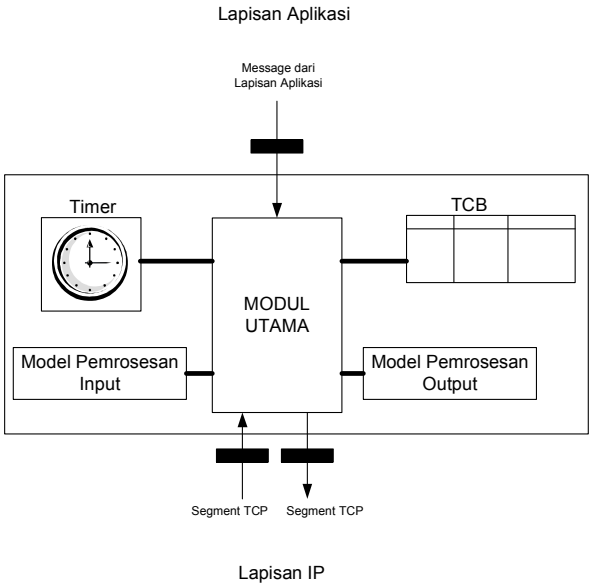
Gambar 2.12 UDP versus IP

Gambar 2.12 menunjukan komunikasi antar *host-to-host* dan *process-to-process* serta perbedaannya. Protokol IP hanya bertanggung jawab mebangun komunikasi antar host dengan host, padahal setelah komunikasi ini terbentuk belum lengkap tanpa disrertai proses yang benar. Adapun kegunaan *UDP* :

1. *UDP* cocok untuk proses yang memerlukan *request-respont communication*.
2. *UDP* yang melakukan proses dengan mekanisme internal *flow control* dan *error control*.
3. *UDP* cocok untuk *multicasting* dan *broadcasting* pada lapisan *transport*.

2.10.2 TCP (*Transmission Control Protocol*)

TCP melakukan layanan *stream* data pada lapisan *transport*, pengiriman *stream* dan penerima *TCP* menggunakan *buffer*. Data yang dilakukan secara *stream* itu berupa segmen-segmen. Beberapa poin yang didapat dalam *TCP* adalah melakukan operasional sebagai berikut



Gambar 2.13 *Disain TCP*

Pada gambar 2.13 menunjukan Enkapsulasi dan dekasulasi (pembungkusan dan pembuka bunkus), *Queuing* (pengantrian), *Multiplexing* dan *demultiplexing*, *Pushing data*.

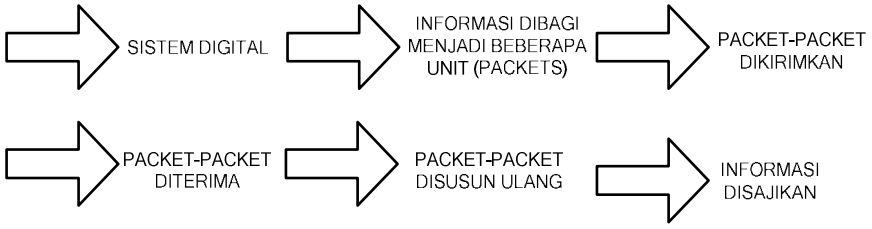
2.11 Jaringan Multimedia

Multimedia dapat diartikan sebagai penggunaan beberapa media yang berbeda untuk menggabungkan dan menyampaikan informasi dalam bentuk text, audio, grafik, animasi, dan video. Istilah multimedia mengacu pada kelas media yang berbeda untuk memakai informasi lalu lintas

multimedia yang mengacu pada tranmisi data pada jaringan komunikasi. Sedangkan multimedia berbasis jaringan adalah sistem ini harus terhubung melalui jaringan yang mempunyai bandwidth yang besar. Perbedaannya adalah adanya *sharing* sistem dan pengaksesan terhadap sumber daya yang sama. Contoh: *video converence* dan *video broadcast* tapi permasalahannya adalah bila bandwidth kecil, maka akan terjadi kemacetan jaringan, delay dan masalah infrastruktur yang belum siap

2.12 Data *Stream*.

Dalam sistem multimedia terdistribusi, data ditransmisikan (*time dependent*) dan terjadi pertukaran informasi



Gambar 2.14 Transmisi Informasi

Pada gambar 2.14 menunjukan Transmisi informasi yang dapat dikategorikan :

1. Berdasarkan mode transmisi

a. *Ansynchronous Trasmission Mode*

1. Komunikasi tanpa batas waktu. Paket mencapai penerima secepat mungkin
2. Paket yang dikirim cepat karena tidak perlu adanya sinkronisasi
3. Informasi untuk *discrete* media dapat ditransmisikan sebagai *ansynchronous data stream*

4. Contoh : transmisi e-mail

b. *Synchronous Transmission Mode*

1. Terdapat batas waktu tunda maksimal untuk setiap packet dari suatu data *stream*
2. Butuh sinkronisasi
3. Penerima butuh *buffer* untuk menyimpan data sementara sambil menunggu paket lengkap

c. *Isochronous Transmission Mode*

1. Terdapat batas waktu tunda maksimal dan minimal
2. Melakukan garansi paket diterima dengan baik
3. Client memberikan informasi kepada server tentang statusnya
4. Membutuhkan *buffer* yang sangat besar

2. Berdasar periode *streaming*

a. *Strongly Periodic Stream*

Interval waktu antara dua packet yang berurutan tetap

Contoh : PCM coded

b. *Weakly Periodic Stream*

Interval waktu antara dua packet yang berurutan dapat dideskripsikan dengan fungsi periodik

c. *Aperiodic Stream*

Interval waktu tidak beraturan

3. Berdasar ukuran packet

a. *Strongly Regular Stream*

Ukuran packet konstan

Contoh: *uncompressed audio/video stream*

b. *Weakly Regular Stream*

Ukuran packet data berubah secara periodik

Contoh: MPEG

c. *Irregular Data Stream*

Ukuran packet data tidak tentu

2.13 Bahasa Pemrograman Delphi

Borland Delphi 7 adalah paket bahasa pemrograman yang bekerja dalam sistem operasi windows. Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan sangat canggih. Berbagai jenis aplikasi dapat dibuat dengan Delphi, termasuk aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web.

Secara umum, kemampuan Delphi adalah menyediakan komponen-komponen dan bahasa pemrograman yang andal, sehingga memungkinkan untuk dapat membuat program aplikasi sesuai dengan keinginan, dengan tampilan dan kemampuan yang canggih.

Untuk mempermudah *programmer* dalam membuat program aplikasi, Delphi menyediakan fasilitas pemrograman yang sangat lengkap. Fasilitas pemrograman tersebut dibagi dalam 2 kelompok, yaitu *object* dan bahasa pemrograman. Secara ringkas, *object* adalah suatu komponen yang mempunyai bentuk fisik dan biasanya dapaat dilihat (visual). *Object* biasanya dipakai untuk melakukan tugas tertentu dan mempunyai batasan-batasan tertentu. Sedangkan

bahasa pemrograman secara singkat dapat disebut sebagai sekumpulan teks yang mempunyai arti tertentu dan disusun dengan aturan tertentu serta untuk menjalankan tugas tertentu. Delphi menggunakan struktur bahasa pemrograman *Object* Pascal yang sudah sangat dikenal di kalangan pemrogram profesional. Gabungan dari *object* dan bahasa pemrograman ini sering disebut sebagai bahasa pemrograman berorientasi *object* atau *Object Oriented Programing* (OOP).

Secara audio delphi telah memberikan komponen-komponen yang mudah dalam membangun aplikasi tersebut misalnya *vaxextended* salah satu fungsinya yaitu mengirimkan data dalam bentuk audio . Dalam grafis pun delphi dapat mengolah data yang membuat tampilan gambar seperti grafik equalizer dengan menggunakan komponen khsus. Secara matematis delphi 7 telah menyediakan operator untuk mengolah data sehingga dalam membuat aplikasinya pun lebih mudah, adapun tingkatan operator dalam delphi 7 sehingga jika muncul fungsi matematis maka akan diolah secara otomatis.

Tabel 2.3 Hirarki Operator Matematis

Operator	Hirarki
@, not	Tertinggi
*, /, div, mod, and, shl, sh	Kedua
+, -, or, xor	Ketiga
=, <>, <, >, <=, >=, in, is	Keempat

Pada Tabel 2.3 menunjukan hirarki operator dalam Delphi 7.