

**ANIMASI PEMBELAJARAN ALGORITMA
AUTOMATIC REPEAT REQUEST (ARQ)
PADA PENGONTROLAN KESALAHAN
PENGIRIMAN DATA**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat
Untuk Memperoleh Gelar Sarjana

Oleh :

**ZULFADLI INDRA
10351022899**



**FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SULTAN SYARIF KASIM RIAU
PEKANBARU
2009**

**ANIMASI PEMBELAJARAN ALGORITMA *AUTOMATIC REPEAT
REQUEST (ARQ)* PADA PENGONTROLAN KESALAHAN
PENGIRIMAN DATA**

**ZULFADLI INDRA
10351022899**

Tanggal Sidang : 09 Desember 2009

Periode Wisuda : Februari, 2010

Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau

ABSTRAK

Dalam pengiriman paket data melalui suatu medium tertentu, akan selalu terjadi adanya perbedaan antara sinyal yang dikirim dengan sinyal yang diterima. Perbedaan tersebut dapat mengakibatkan adanya kesalahan pada paket data yang dikirim sehingga data tersebut tidak dapat dibaca dengan baik oleh penerima. Untuk menghindari terjadinya kesalahan pada paket data yang dikirim maka diperlukan adanya pengontrolan kesalahan (*error control*).

Teknik yang umum digunakan untuk mengontrol kesalahan adalah pendeteksian kesalahan, balasan positif, transmisi ulang setelah waktunya habis, balasan negatif dan retransmisi. Mekanisme dari pengontrolan kesalahan ini disebut sebagai *Automatic Repeat Request (ARQ)*. Tiga metode ARQ yang sudah distandarisasikan adalah *stop-and-wait ARQ*, *go-back-n ARQ* dan *selective-reject ARQ*.

Animasi sebagai media pembelajaran dibuat untuk menampilkan prosedur pengontrolan kesalahan dengan metode *Automatic Repeat Request (ARQ)* yang terdiri dari *stop-and-wait ARQ*, *go-back-n ARQ*, dan *selective-reject ARQ*.

Masing-masing dari ketiga metode ARQ mempunyai kelebihan dan kekurangan dalam mekanisme pengontrolan kesalahan pengiriman data. Pada sisi proses, metode *selective-reject ARQ* lebih efisien dibandingkan dengan metode *stop-and-wait ARQ* dan *go-back-n ARQ*. Tetapi pada sisi penggunaan, metode *go-back-n ARQ* yang lebih banyak dipergunakan.

Kata Kunci : *Automatic repeat request (ARQ), Frame, Pengontrolan Kesalahan*

**LEARNING ALGORITHM ANIMATION AUTOMATIC
REPEAT REQUEST (ARQ) TO CONTROL DATA
TRANSMISSION ERROR**

**ZULFADLI INDRA
10351022899**

Date of Final Exam : December 09th, 2009

Graduation Cremony Priod : February, 2010

*Engineering Departement of Informatic Technology
Faculty of Sciences and Technology
State Islamic University of Sultan Syarif Kasim Riau*

ABSTRACT

In the delivery of data packets through a particular medium, there will always be differences between the signal that is sent with the signal received. This difference can cause the error in the packet of data sent so that data can not be read correctly by the recipient. To avoid the occurrence of error in the packet of data sent and the required error control.

A common technique used to control the error is an error detection, positive response, re-transmission after the time expires, the negative response and retransmissions. The mechanism of controlling this error is called Automatic Repeat Request (ARQ). Three methods that have been standardized ARQ is a stop-and-wait ARQ, go-back-N ARQ and selective-reject ARQ.

Animation as a medium of learning is made to display the error control procedure with the method of Automatic Repeat Reques (ARQ) which consists of a stop-and-wait ARQ, go-back-n ARQ, and selective-reject ARQ

Each of the three ARQ method has advantages and disadvantages of the mechanisms controlling data transmission errors. In the process method of ARQ selective-reject more than the method efficient stop-and-wait ARQ and go-back-n ARQ. But on the usage, method of go-back-N ARQ is more widely used.

Keyword: *Automatic Repeat Request (ARQ), Error Control, frame*

DAFTAR ISI

	Halaman
LEMBAR PERSETUJUAN	ii
LEMBAR PENGESAHAN	iii
LEMBAR HAK ATAS KEKAYAAN INTELEKTUAL.....	iv
LEMBAR PERNYATAAN.....	v
LEMBAR PERSEMBAHAN	vi
ABSTRAK.....	vii
<i>ABSTRACT</i>	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL.....	xvi
DAFTAR LAMPIRAN.....	xvii
BAB I PENDAHULUAN	I-1
1.1 Latar Belakang	I-1
1.2 Rumusan Masalah	I-2
1.3 Tujuan Tugas Akhir	I-3
1.4 Batasan Masalah	I-3
1.5 Sistematika Pembahasan	I-3
BAB II LANDASAN TEORI.....	II-1
2.1 Model Komunikasi.....	II-1
2.2 Terminologi Transmisi.....	II-3
2.3 Gangguan Transmisi	II-3
2.4 Konsep OSI <i>Layer</i>	II-6
2.4.1 Karakteristik Lapisan OSI.....	II-8
2.4.2 Lapisan-lapisan Model OSI	II-9
2.5 <i>Protocol</i>	II-11
2.5.1 TCP (<i>Transmission Control Protocol</i>).....	II-12
2.5.2 Prinsip Kerja TCP (<i>Transmission Control Protocol</i>)	II-15
2.6. Pengontrolan Kesalahan.....	II-16

2.6.1	<i>Stop-and-Wait</i> ARQ.....	II-18
2.6.2	<i>Go-back-N</i> ARQ	II-21
2.6.3	<i>Selective-Reject</i> ARQ.....	II-25
2.7	<i>Clock tick</i>	II-27
2.8	Animasi	II-28
BAB III	METODOLOGI PENELITIAN.....	III-1
3.1	Identifikasi Masalah.....	III-2
3.2	Studi Pustaka.....	III-2
3.3	Analisa	III-2
3.4	Perancangan	III-3
3.5	Implementasi.....	III-3
BAB IV	ANALISIS DAN PERANCANGAN	IV-1
4.1.	Analisis Animasi	IV-1
4.1.1	Analisis Animasi Metode <i>Stop-and-Wait</i> ARQ.....	IV-1
4.1.2	Analisis Animasi Metode <i>Go-back-N</i> ARQ	IV-4
4.1.3	Analisis Animasi Metode <i>Selective-Reject</i> ARQ.....	IV-7
4.1.4	Analisis Data Masukan Menggunakan Metode <i>Stop-and-wait</i> ARQ	IV-9
4.1.5	Analisis Data Masukan Menggunakan Metode <i>Go-back-N</i> ARQ	IV-10
4.1.6	Analisis Data Masukan Menggunakan Metode <i>Selective-Reject</i> ARQ.....	IV-10
4.1.7	Analisis Proses Pengaturan Waktu Animasi.....	IV-11
4.1.8	Analisis Proses Transmisi Frame oleh <i>Transmitter</i>	IV-12
4.1.9	Analisis Proses Pengiriman Frame	IV-13
4.1.10	Analisis Proses Penerimaan Frame oleh <i>Receiver</i>	IV-14
4.1.11	Analisis Proses Transmisi Balasan dari <i>Receiver</i>	IV-15
4.1.12	Analisis Proses Pengiriman Balasan.....	IV-15
4.1.13	Analisis Proses Penerimaan Balasan oleh <i>Transmitter</i>	IV-16
4.1.14	Analisis Data Keluaran (<i>output</i>)	IV-24
4.2	Perancangan Animasi.....	IV-24
4.2.1	<i>Form</i> Utama	IV-25

4.2.2	<i>Form Input</i> Animasi Metode <i>Stop-and-Wait</i> ARQ.....	IV-26
4.2.3	<i>Form</i> Animasi Metode <i>Stop-and-Wait</i> ARQ	IV-27
4.2.4	<i>Form Input</i> Animasi Metode <i>Go-back-N</i> ARQ	IV-28
4.2.5	<i>Form</i> Animasi Metode <i>Go-back-N</i> ARQ.....	IV-30
4.2.6	<i>Form Input</i> Animasi Metode <i>Selective-Reject</i> ARQ.....	IV-32
4.2.7	<i>Form</i> Animasi Metode <i>Selective-Reject</i> ARQ.....	IV-33
4.2.8	<i>Form</i> Metode ARQ	IV-35
4.2.9	<i>Form About</i>	IV-36
BAB V IMPLEMENTASI DAN PENGUJIAN		V-1
5.1	Implementasi	V-1
5.1.1	Lingkungan Implementasi	V-1
5.1.2	Batasan Implementasi	V-1
5.1.3	Implementasi Antar Muka	V-2
5.2	Pengujian	V-7
5.2.1	Contoh <i>input</i> data animasi metode <i>Stop-and-Wait</i> ARQ.....	V-7
5.2.2	Contoh <i>input</i> data animasi metode <i>Go-back-N</i> ARQ	V-9
5.2.3	Contoh <i>input</i> data animasi metode <i>Selective-Reject</i> ARQ.	V-11
BAB VI PENUTUP		VI-1
6.1	Kesimpulan	VI-1
6.2	Saran	VI-2

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR RIWAYAT HIDUP

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dalam pengiriman paket data melalui suatu medium tertentu, akan selalu terjadi adanya perbedaan antara sinyal yang dikirim dengan sinyal yang diterima. Perbedaan tersebut dapat mengakibatkan adanya *error* atau kesalahan pada paket data yang dikirim, sehingga data tersebut tidak dapat dibaca dengan baik oleh penerima. Untuk menghindari terjadinya *error* atau kesalahan pada paket data yang dikirim maka diperlukan adanya deteksi kesalahan (*checking error*), perbaikan kesalahan (*correction error*), dan pengontrolan kesalahan (*control error*).

Didalam komunikasi data, pengontrolan kesalahan (*error control*) berkaitan dengan mekanisme untuk mendeteksi dan memperbaiki kesalahan yang terjadi pada transmisi data dalam bentuk frame. Data yang dikirim sebagai deretan frame, dan frame tiba sesuai dengan perintah yang sama saat dikirim. Masing – masing frame yang ditransmisikan mengalami perubahan dan sejumlah variabel penundaan sebelum mencapai *receiver* (penerima). Kesalahan yang mungkin terjadi adalah hilangnya frame (frame gagal diterima *receiver*) dan rusaknya frame (frame diakui telah tiba, namun beberapa bit mengalami kesalahan, sehingga dibuang oleh *receiver*).

Secara umum, teknik yang paling umum untuk mengontrol kesalahan adalah pendeteksian kesalahan, balasan positif (mengembalikan balasan positif

untuk frame bebas-kesalahan yang diterima dengan baik), transmisi ulang setelah waktunya habis (sumber melakukan retransmisi frame yang belum dibalas setelah beberapa saat tertentu), balasan negatif dan retransmisi. Mekanisme dari pengontrolan kesalahan ini disebut sebagai *Automatic Repeat Request (ARQ)*. Tiga metode ARQ yang sudah distandarisasikan adalah *Stop-and-Wait ARQ*, *Go-Back-N ARQ* dan *Selective-Reject ARQ*.

Pada dasarnya animasi merupakan rangkaian gambar yang membentuk sebuah gerakan yang dapat menjelaskan prosedur dan urutan kejadian. Lowe menyatakan bahwa pemula yang tidak memiliki pengetahuan awal akan cenderung untuk lebih memperhatikan perubahan animasi yang menarik secara perseptual dibandingkan dengan perubahan dalam memahami materi. Pemahaman materi secara perseptual akan lebih mudah dipahami apabila ditranlasikan kedalam bentuk animasi media pembelajaran. Sedangkan pemahaman materi tanpa animasi akan sulit dipahami, karena membutuhkan kemampuan memori pengingat manusia yang cukup tinggi terutama bagi mereka yang digolongkan sebagai *prior knowledge* (pengetahuan awal) terhadap konsep yang akan dijelaskan. (Lowe, 2003)

Berdasarkan penjelasan diatas, maka perlu dibuat suatu model animasi pembelajaran algoritma ARQ (*Automatic Repeat Request*) untuk mengontrol kesalahan pada pengiriman data dalam bentuk frame.

1.2. Rumusan Masalah

Berdasarkan latar belakang masalah, maka dapat dirumuskan masalah yaitu “bagaimana membuat animasi pembelajaran algoritma *Automatic Repeat Request* (ARQ) pada pengontrolan kesalahan”

1.3. Batasan Masalah

Adapun batasan masalah dalam tugas akhir ini adalah sebagai berikut :

1. Parameter inputan dalam animasi adalah waktu transmisi per frame, waktu transmisi balasan, interval waktu *time out*, jumlah frame yang ditransmisikan, dan kecepatan animasi.
2. Pada metode *Go-Back-N* ARQ dan *Selective-Reject* ARQ, menampilkan animasi *sliding window flow control* baik dari perspektif pengirim maupun dari perspektif penerima.

1.4. Tujuan

Tujuan yang ingin dicapai dalam pembuatan tugas akhir ini adalah membuat suatu animasi pembelajaran untuk menampilkan proses kerja algoritma ARQ pada pengontrolan kesalahan.

1.5. Sistematika Penulisan

Sistematika penulisan dalam penyusunan laporan tugas akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan dasar-dasar dari penulisan laporan tugas akhir, yang terdiri dari latar belakang, rumusan masalah, batasan masalah, tujuan, serta sistematika penulisan laporan tugas akhir.

BAB II LANDASAN TEORI

Bab ini membahas teori-teori yang berhubungan dengan topik penelitian, yang terdiri dari model komunikasi, konsep osi *layer*, data *link control*, animasi, dan teori lainnya yang berhubungan dengan topik tugas akhir ini.

BAB III METODOLOGI PENELITIAN

Bab ini berisikan pedoman tentang tahapan-tahapan yang dilakukan dalam pelaksanaan penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang telah ditentukan sebelumnya.

BAB IV ANALISIS DAN PERANCANGAN

Bab ini membahas tentang hasil analisis animasi pembelajaran dengan menggunakan algoritma ARQ, dan perancangan perangkat lunak yang akan dibuat.

BAB V IMPLEMENTASI DAN PENGUJIAN

Bab ini membahas implementasi terhadap animasi yang telah dibuat yaitu animasi pembelajaran algoritma ARQ pada pengontrolan kesalahan dalam proses pengiriman data dalam bentuk frame.

BAB VI PENUTUP

Bab ini berisi kesimpulan dan saran yang dihasilkan dari pembahasan tentang animasi pembelajaran algoritma ARQ pada pengontrolan kesalahan.

BAB II

LANDASAN TEORI

2.1. Model Komunikasi

Kegunaan dasar dari sistem komunikasi adalah menjalankan pertukaran data antara 2 pihak. Contohnya adalah pertukaran sinyal-sinyal suara antara 2 telepon pada suatu jaringan yang sama (Stallings,2001). Elemen-elemen kunci model komunikasi adalah:

1. *Source* (Sumber)

Alat ini membangkitkan data sehingga dapat ditransmisikan, contoh: telepon dan PC (Personal Computer).

2. *Transmitter* (Pengirim)

Biasanya data yang dibangkitkan dari sistem sumber tidak ditransmisikan secara langsung dalam bentuk aslinya. Sebuah *transmitter* cukup memindah dan menandai informasi dengan cara yang sama seperti menghasilkan sinyal-sinyal elektro-magnetik yang dapat ditransmisikan melewati beberapa sistem transmisi berurutan.

3. *Transmission System* (Sistem Transmisi)

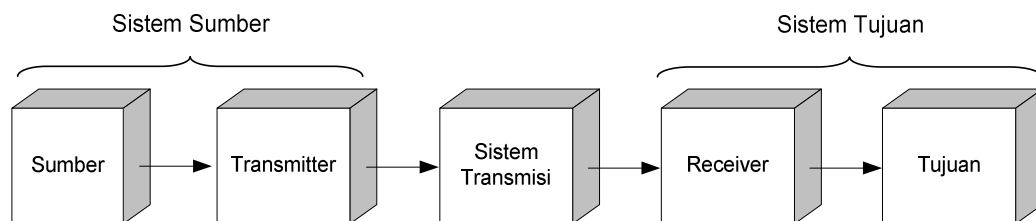
Berupa jalur transmisi tunggal (*single transmission line*) atau jaringan kompleks (*complex network*) yang menghubungkan antara sumber dengan tujuan (*destination*).

4. *Receiver* (Penerima)

Receiver menerima sinyal dari sistem transmisi dan menggabungkannya ke dalam bentuk tertentu yang dapat ditangkap oleh tujuan.

5. *Destination* (Tujuan)

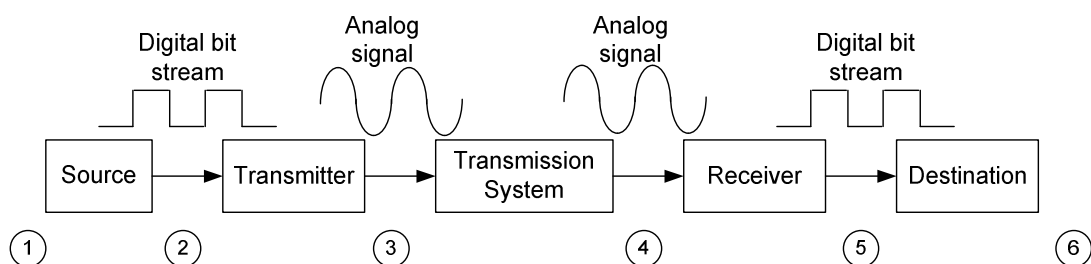
Menangkap data yang dihasilkan oleh *receiver*.



Gambar 2.1. Model Komunikasi

Source (sumber) dan *transmitter* (pengirim) merupakan bagian dari sistem sumber, sedangkan *receiver* (penerima) dan *destination* (tujuan) merupakan bagian dari sistem tujuan.

Proses transmisi / pengiriman data secara terperinci dapat dilihat pada gambar di bawah ini,



Gambar 2.2. Rincian Proses Transmisi Data

Keterangan :

1. Informasi yang di-*input*.
2. Data yang di-*input*.
3. Signal yang akan ditransmisikan berupa signal analog.
4. Signal yang diterima berupa signal analog.

5. Data *output*.
6. Informasi output.

2.2. Terminologi Transmisi

Transmisi data terjadi di antara *transmitter* dan *receiver* melalui beberapa media transmisi. Media transmisi dapat digolongkan sebagai *guided* atau *unguided*. Pada kedua hal itu, komunikasi berada dalam bentuk gelombang elektromagnetik. Dengan *guided* media, gelombang dikendalikan sepanjang jalur fisik. Contoh-contoh *guided* media adalah *twisted pair*, *coaxial kabel*, serta *fiber optik*. *Unguided* media menyediakan alat untuk mentransmisikan gelombang-gelombang elektromagnetik namun tidak mengendalikannya, contohnya adalah perambatan (*propagation*) di udara dan laut.

Sebuah transmisi dapat berupa *simplex*, *half duplex*, atau *full duplex*. Pada transmisi *simplex*, sinyal ditransmisikan hanya pada satu *direction* (arah), satu station sebagai *transmitter* dan lainnya sebagai *receiver*. Pada operasi *half-duplex*, kedua station dapat mentransmisikan, namun hanya satu station pada saat yang sama. Sedangkan pada operasi *full duplex*, kedua station bisa mentransmisikan data secara bersamaan. (Stallings, 2001)

2.3. Gangguan Transmisi

Dalam sistem komunikasi, sinyal yang diterima kemungkinan berbeda dengan sinyal yang ditransmisikan dikarenakan adanya berbagai gangguan transmisi. Bagi analog signal, gangguan ini dapat menurunkan kualitas sinyal. Sedangkan bagi digital signal, akan muncul bit *error* pada *receiver*.

Secara umum, gangguan transmisi yang paling signifikan adalah:

a. *Atenuasi*

Kekuatan sinyal akan melemah karena jarak yang jauh melalui medium transmisi apapun. Ada 3 pertimbangan untuk perancang transmisi:

1. Sinyal yang diterima harus mempunyai kekuatan yang cukup sehingga penerima dapat mendeteksi dan mengartikan sinyal tersebut.
2. Sinyal harus mencapai suatu level yang lebih tinggi dari *noise* agar dapat diterima tanpa kesalahan.
3. *Atenuasi* adalah suatu fungsi dari frekuensi.

b. Distorsi Penundaan

Distorsi penundaan (*delay distortion*) terjadi akibat kecepatan sinyal yang melalui medium yang berbeda-beda sehingga sampai pada penerima dengan waktu yang berbeda. Hal ini merupakan hal kritis bagi data digital yang dibentuk dari sinyal-sinyal dengan frekuensi-frekuensi yang berbeda sehingga menyebabkan *intersymbol interference*.

c. Derau (*Noise*)

Untuk suatu peristiwa pentransmisi data, sinyal yang diterima akan berisikan sinyal-sinyal yang ditransmisikan, dimodifikasi oleh berbagai distorsi yang terjadi melalui sistem transmisi, plus sinyal-sinyal tambahan yang tidak diinginkan yang diselipkan di suatu tempat di antara transmisi dan penerimaan. Berikutnya, sinyal-sinyal yang tidak diharapkan tersebut disebut sebagai derau. Derau dapat dibagi menjadi empat kategori, yaitu:

1. Derau suhu (*Thermal Noise*)

Merupakan suatu gejolak thermal elektron. Muncul di semua perangkat elektronik dan media transmisi serta merupakan fungsi temperatur. Derau suhu secara keseluruhan disebarkan sepanjang spektrum frekuensi dan sering juga disebut sebagai *white noise*. Derau suhu tidak dapat dihilangkan dan karena itu menempatkan suatu batas atas pada unjuk kerja sistem komunikasi.

2. Derau Intermodulasi (*Intermodulation Noise*)

Disebabkan sinyal pada frekuensi-frekuensi yang berbeda tersebar pada medium transmisi yang sama sehingga menghasilkan sinyal pada suatu frekuensi yang merupakan penjumlahan dari frekuensi-frekuensi asalnya. Hal ini timbul karena ketidak linearan *transmitter*, *receiver* atau sistem transmisi.

3. *Crosstalk*

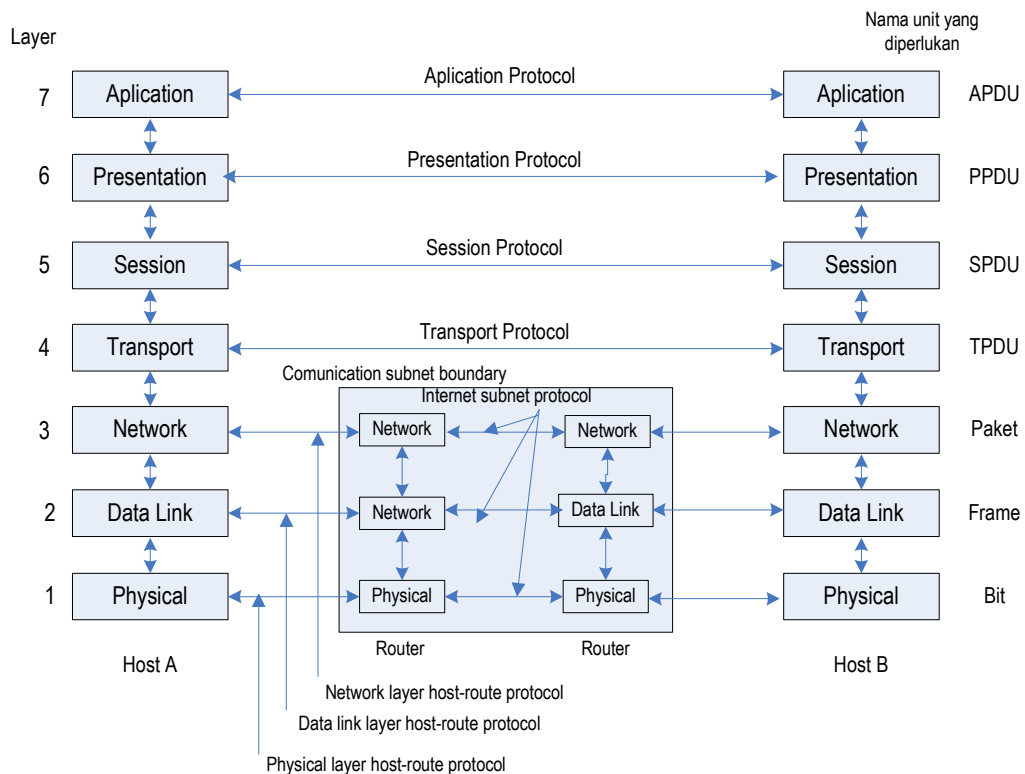
Crosstalk dialami oleh siapapun yang saat menggunakan telepon, terdengar percakapan lain; ini merupakan kopel yang tidak diharapkan yang terjadi di antara sinyal. Dapat pula terjadi karena kopel elektrik di antara *twisted pair* yang berdekatan. *Crosstalk* dapat pula terjadi bila sinyal-sinyal yang tidak diharapkan tersebut disebarkan melalui antena gelombang mikro, meskipun antena pengarah dipergunakan, namun energi gelombang mikro tersebar luas selama proses propagasi.

4. Derau *impuls*

Derau *impuls* umumnya hanyalah gangguan kecil bagi data analog. Sebagai contoh, transmisi suara dapat diganggu oleh bunyi klik dan gemerisik tanpa mengurangi kejelasannya. Bagaimanapun juga, derau impuls juga merupakan sumber utama terjadinya *error* dalam komunikasi data digital. Gangguan ini kadang-kadang cukup memadai untuk mengubah 1 menjadi 0 atau 0 menjadi 1. (Stallings,2003)

2.4. Konsep OSI Layer

Model referensi OSI (*Open System Interconnection*) menggambarkan bagaimana informasi dari suatu software aplikasi di sebuah komputer berpindah melewati sebuah media jaringan ke suatu software aplikasi di komputer lain. Model referensi OSI secara konseptual terbagi ke dalam 7 lapisan, dimana masing-masing lapisan memiliki fungsi jaringan yang spesifik. Model ini diciptakan berdasarkan sebuah proposal yang dibuat oleh the *International Standards Organization* (ISO) sebagai langkah awal menuju standarisasi protokol internasional yang digunakan pada berbagai layer. Model ini disebut ISO OSI *Reference* model karena model ini ditujukan bagi pengkoneksian open sistem. Open sistem dapat diartikan sebagai suatu sistem yang terbuka untuk berkomunikasi dengan sistem - sistem lainnya.



Gambar 2.3. Model Referensi OSI

Model OSI memiliki tujuh layer. Prinsip-prinsip yang digunakan bagi ketujuh layer tersebut adalah :

1. Sebuah layer harus dibuat bila diperlukan tingkat abstraksi yang berbeda.
2. Setiap layer harus memiliki fungsi-fungsi tertentu.
3. Fungsi setiap layer harus dipilih dengan teliti sesuai dengan ketentuan standar protocol internasional.
4. Batas-batas layer diusahakan agar meminimalkan aliran informasi yang melewati *interface*.
5. Jumlah layer harus cukup banyak, sehingga fungsi-fungsi yang berbeda tidak perlu disatukan dalam satu layer diluar keperluannya. Akan tetapi

jumlah layer juga harus diusahakan sesedikit mungkin sehingga arsitektur jaringan tidak menjadi sulit dipakai.

2.4.1. Karakteristik Lapisan OSI

Lapisan atas dari model OSI berurusan dengan persoalan aplikasi dan pada umumnya diimplementasi hanya pada software. Lapisan tertinggi (*aplication layer*) adalah lapisan penutup sebelum ke pengguna (*user*).

Lapisan bawah dari model OSI mengendalikan persoalan transport data. Lapisan fisik dan lapisan data link diimplementasikan ke dalam hardware dan software. Lapisan - lapisan bawah yang lain pada umumnya hanya diimplementasikan dalam software. Lapisan terbawah, yaitu lapisan fisik adalah lapisan penutup bagi media jaringan fisik (misalnya jaringan kabel), dan sebagai penanggung jawab bagi penempatan informasi pada media jaringan. Tabel berikut ini menampilkan pemisahan kedua lapisan tersebut pada lapisan-lapisan model OSI.

Tabel 2.1. Pemisahan lapisan atas dan lapisan bawah pada model OSI

Application	Application	Lapisan Atas
Presentation		
Session		
Transport	Data Transport	Lapisan Bawah
Network		
Data Link		
Physical		

2.4.2. Lapisan-lapisan Model OSI

Teknik pada model referensi OSI adalah menggunakan teknik *layer* (lapisan) dimana setiap lapisan dibedakan menurut fungsi dan proses yang dilakukan. Fungsi-fungsi yang mirip harus dijadikan satu lapisan sehingga tidak tercipta banyak lapisan yang akan mengakibatkan transmisi data menjadi lambat dan tidak efektif. Model referensi OSI ini didefinisikan menjadi tujuh lapisan protokol komunikasi, yaitu:

a. *Physical Layer*

Physical Layer berfungsi dalam pengiriman raw bit ke channel komunikasi. Lapisan ini merupakan lapisan pertama dan bertugas untuk mengatur sinkronisasi pengiriman dan penerimaan data, spesifikasi mekanis dan elektrik, membangun dan memutuskan hubungan komunikasi.

b. *Data Link Layer*

Tugas utama data *link layer* adalah sebagai fasilitas transmisi raw data dan mentransformasi data tersebut ke saluran yang bebas dari kesalahan transmisi. Fungsi dari lapisan ini antara lain : Pertama, memecah data atau informasi menjadi beberapa frame tertentu yang dilengkapi dengan bit-bit alamat pengirim dan penerima. Kedua, mendeteksi kesalahan yang mungkin terjadi saat proses transmisi berlangsung. Ketiga, pada sisi penerima lapisan ini berfungsi untuk menggabungkan kembali bit-bit yang diterima. Terjadinya *noise* pada saluran dapat merusak frame. Dalam hal ini, perangkat lunak data link layer pada mesin sumber dapat mengirim kembali frame yang rusak tersebut. Akan tetapi transmisi frame sama secara

berulang-ulang bisa menimbulkan duplikasi frame. Frame duplikat perlu dikirim apabila *acknowledgement* frame dari penerima yang dikembalikan ke pengirim telah hilang. Tergantung pada layer inilah untuk mengatasi masalah-masalah yang disebabkan rusaknya, hilangnya dan duplikasi frame.

c. *Network Layer*

Lapisan ini merupakan lapisan ketiga dari model OSI. Lapisan ini berfungsi untuk memberikan layanan pengiriman data dengan menentukan rute pengiriman dan mengendalikannya sehingga data dapat sampai ke tujuan. Selain itu data atau informasi yang berupa pesan (*message*) akan dibagi-bagi dalam bentuk paket-paket data yang dilengkapi dengan berbagai *header* tertentu pada setiap paket data tersebut.

d. *Transport Layer*

Merupakan lapisan ke-4 dari model OSI. Fungsinya adalah memberikan layanan dalam hal *error recovery* dan data *flow control* serta mencari rute kosong untuk proses transmisi data dan informasi. Adapun contoh protokol yang digunakan antara lain TP-NBS dan TCP (*Transmission Control Protocol*). TCP secara spesifik dirancang untuk menyediakan aliran data dari terminal yang satu ke terminal yang lain dalam suatu inter jaringan.

e. *Session Layer*

Lapisan session merupakan lapisan ke-5 dari model OSI. Lapisan ini bertugas untuk menyediakan sarana pembangunan hubungan dan pengontrolan terhadap kerjasama antar komputer atau program aplikasi yang

sedang berkomunikasi. Dalam beberapa standar protokol jaringan, lapisan Session dan lapisan Transport digabung menjadi satu lapisan.

f. *Presentation Layer*

Merupakan lapisan ke-6 dari model OSI. Lapisan ini mengatasi masalah perbedaan format penyajian data dengan cara mengonversikan *syntax* data yang dikirim agar dapat dimengerti oleh penerima. Lapisan ini juga menyediakan fasilitas untuk melakukan kompresi dan enkripsi-dekripsi data agar keamanan data terjamin.

g. *Application Layer*

Merupakan lapisan paling atas atau lapisan ke-7 dari model OSI. Bertugas untuk mengatur interaksi antara pengguna komputer dengan program aplikasi yang dipakai. Fungsi *application layer* lainnya adalah pemindahan *file*. Sistem *file* yang satu dengan yang lainnya memiliki konvensi penamaan yang berbeda, cara menyatakan baris-baris teks yang berbeda, dan sebagainya. Perpindahan *file* dari sebuah sistem ke sistem lainnya yang berbeda memerlukan penanganan untuk mengatasi adanya ketidakkompatibelan.

2.5. *Protocol*

Protokol merupakan sekumpulan aturan yang mendefinisikan beberapa fungsi dalam proses transmisi data yang harus dipenuhi oleh pengirim dan penerima agar suatu sesi komunikasi data dapat berlangsung dengan baik dan benar. Selain itu *protocol* juga merupakan sekumpulan aturan untuk memecahkan

masalah-masalah yang khusus yang terjadi antar alat-alat komunikasi agar proses transmisi data dapat terjadi dengan baik dan benar.

Secara umum *protocol* berfungsi untuk membangun hubungan antara pengirim dan penerima serta menyalurkan informasi dengan keakuratan yang tinggi. Secara rinci fungsi protokol adalah sebagai berikut :

- a. *Fragmentasi* dan *reassembly*, yaitu membagi-bagi berita dalam bentuk paket-paket pada saat komputer mengirim data dan menggabungkannya lagi setelah data tersebut diterima.
- b. *Encapsulation*, melengkapi paket-paket dengan *addres*, kode koreksi.
- c. *Connection control*, yaitu membangun hubungan komunikasi, melakukan transmisi data, dan mengakhiri hubungan (*connection termination*)
- d. *Flow control*, fungsi *protocol* sebagai pengatur perjalanan data
- e. *Error control*, fungsi *protocol* sebagai pengontrol terjadinya kesalahan dalam komunikasi data.
- f. *Transmission service*, fungsi *protocol* sebagai pemberi pelayanan komunikasi data khususnya yang berkaitan dengan prioritas dan keamanan data.

2.5.1. TCP (*Transmission Control Protocol*)

TCP adalah protokol yang memungkinkan program-program aplikasi untuk mengakses atau menggunakan layanan komunikasi bersifat *connection-oriented*. TCP mampu memberikan jasa pengiriman yang dapat diandalkan (*reliable*) sekaligus bersifat *flow-controlled*. Sifat *flow-controlled* ini memungkinkan peralatan-peralatan jaringan yang berkecepatan rendah (*slower-*

speed network devices) dapat berhubungan dengan peralatan-peralatan jaringan yang berkecepatan tinggi (*higher-speed network devices*). Sifat-sifat dari *protocol* TCP adalah :

a. *Connection-oriented*

Suatu arsitektur atau mekanisme komunikasi data di mana dua perangkat yang akan saling berkomunikasi diharuskan untuk membuat sebuah sesi (*session*) terlebih dahulu. Ketika komunikasi telah selesai, *session* tersebut akan berakhir. Hal inilah yang terjadi dalam komunikasi menggunakan telepon, sebuah koneksi harus tersedia dan terjadi terlebih dahulu sebelum telepon yang dituju dapat digunakan untuk berkomunikasi dengan telepon yang digunakan untuk memanggil.

b. *Reliable* (keandalan)

Keandalan yang dimiliki oleh *protocol* ini disebabkan karena beberapa mekanisme. Mekanisme tersebut adalah:

1. *Checksum*: semua segmen TCP membawa *checksum* yang akan digunakan oleh si penerima (*receiver device*) untuk mengecek adanya *error* baik itu *error* pada data atau pada *header* milik TCP itu sendiri.
2. *Duplicate Data Detection*: kemampuan TCP untuk menjaga setiap *byte* yang diterima agar *byte-byte* tersebut tidak mengalami penggandaan.
3. *Retransmission*: kemampuan TCP untuk mengimplementasikan skema pengiriman ulang untuk data kiriman yang rusak atau hilang.
4. *Sequencing*: kemampuan TCP untuk menyusun segmen-segmen data yang telah diterimanya. Hal ini akan membuat TCP mampu

mengirimkan kembali data tersebut kepada suatu aplikasi dengan susunan yang benar.

5. *Timers*: TCP menggunakan dua timer sekaligus dalam pengiriman data. Dua timer tersebut yakni timer statik dan timer dinamis. *Protocol* yang menjadi pengirim akan menunggu si penerima dalam periode waktu tertentu untuk sebuah “*acknowledgement*”. Jika timer telah habis masa periodenya, si pengirim dapat mengirim kembali (*retransmit*) segment yang akan dikirim.

c. *Stream data transfer*

TCP akan mengelompokkan *byte-byte* yang sebelumnya tidak terstruktur ke dalam bentuk segmen untuk kemudian dikirimkan ke IP. Layanan ini memberikan keuntungan bagi aplikasi-aplikasi karena mereka tidak perlu lagi membuat blok-blok data.

d. *Efficient flow control*

Ketika mengirim ulang *acknowledgement* ke alamat asal, proses TCP yang menerima mengindikasikan nomor urutan yang bisa diterimanya tanpa harus meng-*over flow buffer* internal miliknya.

- e. *Full-duplex operation*, TCP bisa mengirim dan menerima dalam waktu yang bersamaan.

- f. *Multiplexing*, Komunikasi antar *upper-layer* yang terjadi secara simultan bisa dimultiplexikan melalui satu koneksi tunggal.

2.5.2. Prinsip kerja TCP (*Transmission Control Protocol*)

TCP mempunyai prinsip kerja seperti "*virtual circuit*" pada jaringan telepon. TCP lebih mementingkan tata-cara dan keandalan dalam pengiriman data antara dua komputer dalam jaringan. TCP tidak peduli dengan apa-apa yang dikerjakan oleh IP, yang penting adalah hubungan komunikasi antara dua komputer berjalan dengan baik. Dalam hal ini, TCP mengatur bagaimana cara membuka hubungan komunikasi, jenis aplikasi apa yang akan dilakukan dalam komunikasi tersebut (misalnya mengirim *e-mail*, transfer *file*). Di samping itu, juga mendeteksi dan mengoreksi jika ada kesalahan data. TCP mengatur seluruh proses koneksi antara satu komputer dengan komputer yang lain dalam sebuah jaringan komputer.

TCP (*transmission transfer protocol*) berperan didalam memperbaiki pengiriman data yang benar dari suatu klien ke *server*. Data dapat hilang di tengah-tengah jaringan. TCP dapat mendeteksi *error* atau data yang hilang dan kemudian melakukan transmisi ulang sampai data diterima dengan benar dan lengkap.

Berbeda dengan IP yang mengandalkan mekanisme *connectionless*, pada TCP mekanisme hubungan adalah *connection oriented*. Dalam hal ini, hubungan secara logik akan dibangun oleh TCP antara satu komputer dengan komputer yang lain. Dalam waktu yang ditentukan komputer yang sedang berhubungan harus mengirimkan data atau *acknowledge* agar hubungan tetap berlangsung. Jika hal ini tidak sanggup dilakukan maka dapat diasumsikan bahwa komputer yang sedang

berhubungan dengan kita mengalami gangguan dan hubungan secara logik dapat diputus.

TCP mengatur *multiplexing* dari data yang dikirim atau diterima oleh sebuah komputer. Adanya identifikasi pada TCP *header* memungkinkan *multiplexing* dilakukan. Hal ini memungkinkan sebuah komputer melakukan beberapa hubungan TCP secara logik. Bentuk hubungan adalah *full duplex*, hal ini memungkinkan dua buah komputer saling berbicara dalam waktu bersamaan tanpa harus bergantian menggunakan kanal komunikasi. Untuk mengatasi saturasi (*congestion*) pada kanal komunikasi, pada header TCP dilengkapi informasi tentang *flow control*.

2.6. Pengontrolan Kesalahan

Dalam penggunaan data *link control*, ada beberapa persyaratan yang diperlukan untuk mengefektifkan komunikasi data antara dua stasiun *transmitter* dan *receiver* adalah:

a. Sinkronisasi Frame

Blok data dalam jumlah besar akan dipecah-pecah oleh sumber menjadi blok-blok yang lebih kecil yang disebut frame. Permulaan dan ujung setiap frame harus nampak jelas.

b. Flow Control

Station pengirim tidak boleh mengirim frame pada rate yang lebih cepat dibanding rate station penerima dalam menerima frame-frame tersebut.

c. Pengontrolan Kesalahan

Kesalahan-kesalahan bit diakibatkan oleh sistem transmisi yang harus diperbaiki.

d. Pengalamatan

Pada jalur multipoin, seperti *Local Area Network (LAN)*, identitas dua station yang berkomunikasi harus ditentukan dengan jelas.

e. Kontrol data pada jalur yang sama

Biasanya tidak diharapkan memiliki jalur komunikasi yang terpisah secara fisik untuk mengontrol informasi. Karenanya, *receiver* harus mampu membedakan informasi kontrol dari data yang sedang ditransmisikan.

f. Manajemen Jalur

Permulaan, pemeliharaan dan penghentian pertukaran data memerlukan koordinasi dan kerjasama yang baik di antara station. Karena itu diperlukan suatu prosedur manajemen untuk pertukaran ini.

Pengontrolan kesalahan berkaitan dengan mekanisme untuk mendeteksi dan memperbaiki kesalahan yang terjadi pada penransmisian frame. Data dikirim sebagai deretan frame, frame tiba sesuai dengan perintah yang sama saat dikirim, dan masing-masing frame yang ditransmisikan mengalami perubahan dan sejumlah *variabel* penundaaan sebelum mencapai penerima. Kemungkinan terdapat dua jenis kesalahan, yakni:

a. Hilangnya frame

Frame gagal mencapai sisi yang lain. Sebagai contoh, derau yang kuat bisa merusak frame dan *receiver* tidak menyadari bahwa frame telah ditransmisi.

b. Kerusakan frame

Frame diakui telah tiba, namun beberapa bit mengalami kesalahan (sudah berubah selama transmisi).

Teknik yang paling umum untuk mengontrol kesalahan didasarkan atas beberapa atau seluruh unsur berikut ini:

1. Pendeteksian kesalahan

2. Balasan positif

Bertujuan untuk mengembalikan balasan positif untuk frame bebas-kesalahan yang diterima dengan baik.

3. Retransmisi setelah waktunya habis

Sumber melakukan retransmisi frame yang belum dibalas setelah beberapa saat tertentu.

4. Balasan negatif dan retransmisi

Bertujuan untuk mengembalikan balasan negatif kepada frame yang dideteksi mengalami kesalahan. Sumber melakukan retransmisi terhadap frame itu.

Secara keseluruhan, mekanisme ini disebut sebagai *Automatic Repeat Request* (ARQ), Tiga metode ARQ yang sudah distandarisasikan adalah: *stop-and-wait* ARQ, *go-back-n* ARQ dan *selective-reject* ARQ. (Stallings,2001)

2.6.1. Stop-and-Wait ARQ

Stop-and-Wait ARQ didasarkan atas teknik kontrol arus *stop-and-wait flow control*. Station sumber mentransmisikan sebuah frame tunggal dan kemudian harus menunggu balasan (ACK). Tidak ada data yang dikirim sampai

jawaban dari station tujuan tiba di stasiun sumber. Ada dua jenis kesalahan yang dapat terjadi, yaitu:

- a. Frame yang tiba di tujuan mengalami kerusakan atau frame hilang.

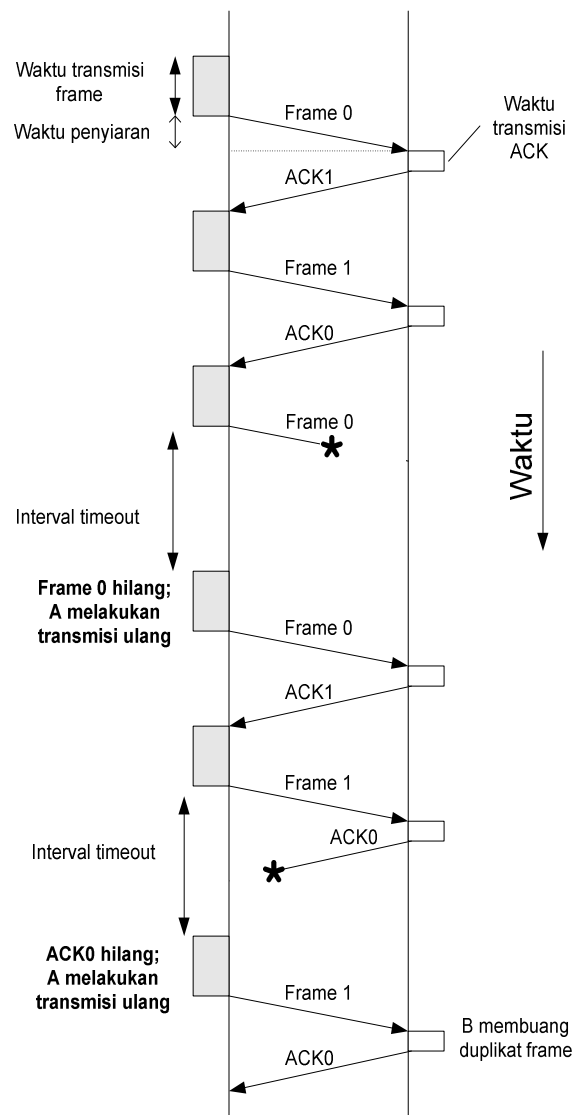
Apabila *receiver* mendeteksi adanya kerusakan pada frame, maka *receiver* akan mengirimkan balasan negatif (NACK) terhadap nomor frame yang rusak dan *transmitter* akan mengirimkan ulang frame tersebut. Untuk kasus frame hilang di tengah perjalanan, station sumber dilengkapi sebuah pencatat waktu. Setelah frame ditransmisikan, station sumber menunggu balasan. Bila tidak ada balasan yang diterima sampai waktu yang ditentukan pencatat waktu habis, maka station sumber akan mengirimkan frame yang sama kembali.

- b. Kerusakan pada balasan positif (ACK) dan balasan negatif (NACK)

Sebagai ilustrasi, station A mengirim sebuah frame dan diterima dengan baik oleh station B, yang kemudian memberi balasan ACK. Namun ACK mengalami kerusakan dan tidak diakui oleh A. Mungkin waktu tanggap untuk memberikan respons yang diberikan stasiun A telah habis dan stasiun A kemudian kembali mengirimkan frame yang sama. Duplikat frame ini tiba dan diterima oleh B. Dengan begitu, B menerima dua duplikat frame yang sama seolah-olah keduanya merupakan frame yang berbeda. Untuk mengatasi problem ini, frame bergantian diberi label 0 atau 1, dan balasan positifnya dalam bentuk ACK0 dan ACK1. Sesuai dengan aturan jendela penggeseran, ACK0 membalas penerimaan frame bernomor 1 dan menunjukkan bahwa *receiver* siap untuk menerima frame bernomor 0.

Untuk kerusakan pada balasan negatif (NACK), *transmitter* mengirimkan frame yang sama (sama seperti kasus frame hilang).

Kelebihan utama *stop-and-wait* ARQ adalah kesederhaannya, sedangkan kekurangan utamanya adalah mekanisme yang dijalankan tidak efisien. Oleh karena itu, teknik kontrol arus jendela penggeseran dapat diadaptasikan agar diperoleh penggunaan jalur yang lebih efisien lagi, seperti yang diimplementasikan pada metode *go-back-n* ARQ dan *selective-reject* ARQ.

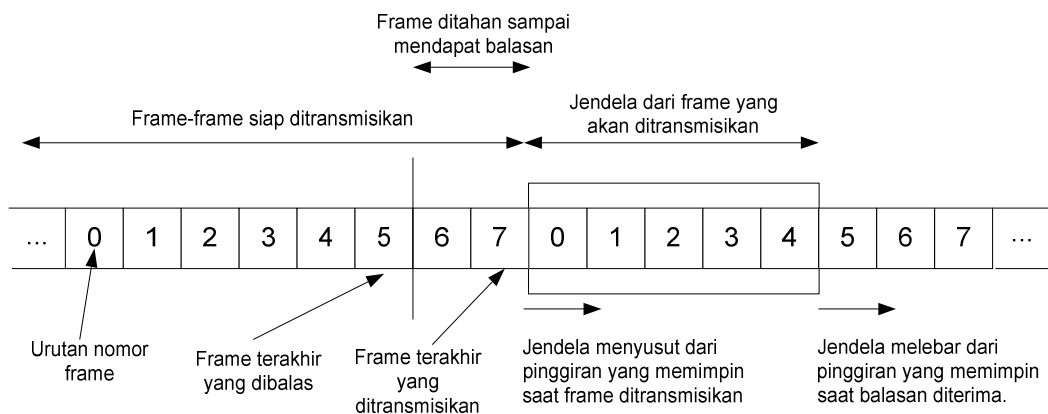


Gambar 2.6. Prosedur *Stop-and-Wait* ARQ

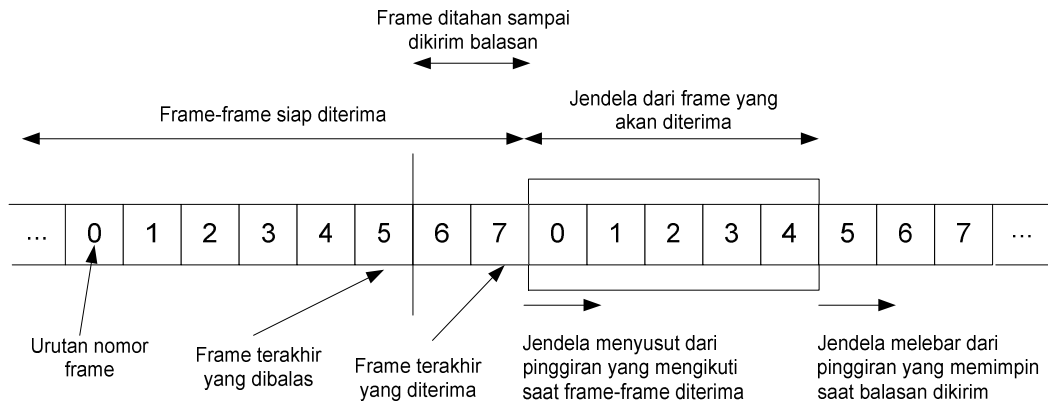
2.6.2. Go-back-N ARQ

Go-back-N ARQ merupakan bentuk pengontrolan kesalahan yang didasarkan atas teknik kontrol arus jendela penggeseran. Dalam metode ini, station sumber bisa mengirim deretan frame yang diurutkan berdasarkan suatu modulo bilangan. Jumlah frame balasan yang ada ditentukan oleh ukuran jendela, menggunakan teknik kontrol arus jendela penggeseran (*Sliding Window Flow Control*)

Sliding window flow control memberikan nomor pada frame-frame yang akan ditransmisikan. Sebagai contoh, untuk bidang 3-bit, urutan nomornya berkisar dari 0 sampai 7. Jadi frame-frame diberi nomor modulo 8, maksudnya setelah nomor urut 7, nomor berikutnya adalah 0. Umumnya, untuk bidang k-bit kisaran urutan nomornya adalah 0 sampai 2^{k-1} dan frame-frame-nya diberi nomor modulo 2^k .



Gambar 2.4. Jendela penggeseran menurut perspektif pengirim



Gambar 2.5. Jendela penggeseran menurut perspektif penerima

Bila tidak terjadi suatu kesalahan, station tujuan akan membalas ($RR = Receive Ready$) frame yang datang seperti biasa. Bila station tujuan mendeteksi suatu kesalahan pada sebuah frame, station tujuan mengirim balasan negatif ($REJ = Reject$) untuk frame tersebut. Station tujuan kemudian membuang frame itu dan semua frame-frame yang nantinya akan datang sampai frame yang mengalami kesalahan diterima dengan benar. Jadi, station sumber, bila menerima REJ , harus melakukan retransmisi terhadap frame yang mengalami kesalahan tersebut plus semua frame pengganti yang ditransmisikan sementara.

Sebagai contoh, station A mengirim frame ke station B. Setelah setiap transmisi dilakukan, A menyusun pencatat waktu balasan untuk frame yang baru saja ditransmisi. Anggap saja bahwa B sebelumnya berhasil menerima frame ($i - 1$) dan A baru saja mentransmisikan frame i . Teknik *go-back-N* mempertimbangkan kemungkinan-kemungkinan berikut ini:

1. Rusaknya frame

Bila frame yang diterima *invalid* (misalnya, B mendeteksi adanya kesalahan), B membuang frame dan tidak melakukan tindakan apa-apa.

Dalam hal ini, ada dua subkasus, yakni:

- a. Di dalam periode waktu yang memungkinkan, A berturut-turut mengirim frame $(i + 1)$. B menerima frame $(i + 1)$ yang tidak beres dan mengirim REJ i . A harus melakukan retransmisi terhadap frame i dan semua frame urutannya.
- b. A tidak segera mengirim frame-frame tambahan. B tidak menerima apa-apa serta tidak mengembalikan RR maupun REJ. Bila pewaktu A habis, A mentransmisikan frame RR yang memuat bit disebut dengan bit P, yang disusun berdasarkan 1. B menerjemahkan frame RR dengan bit P dari 1 sebagai perintah yang harus dijawab dengan jalan mengirimkan RR, menunjukkan frame berikutnya yang diharapkan, yang berupa frame i . Bila A menerima RR, ia kembali mentransmisikan frame i .

2. Rusaknya RR

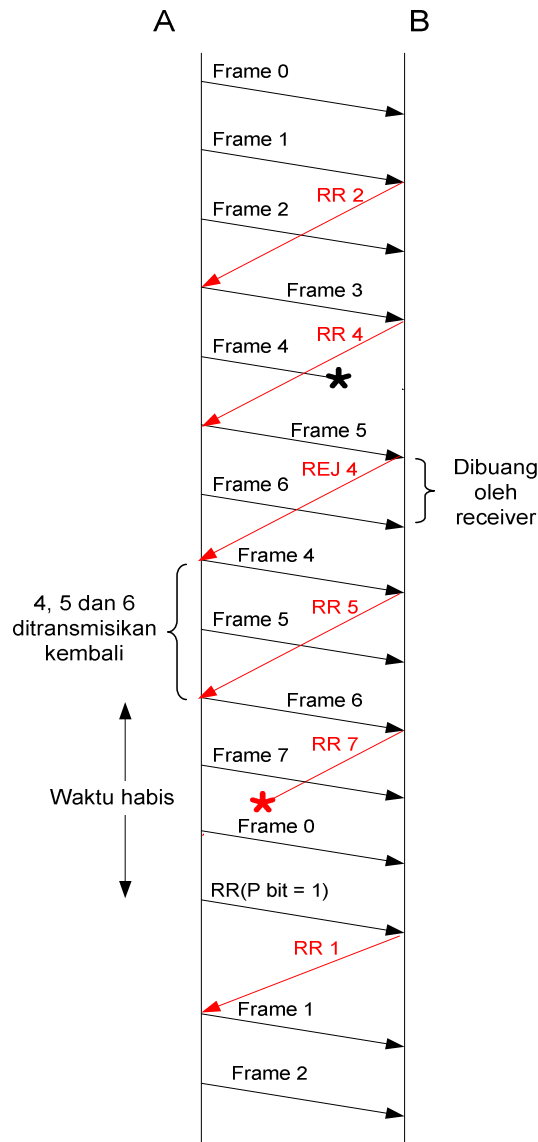
Terdapat dua subkasus, yakni:

- a. B menerima frame i dan mengirim $RR(i+1)$, yang hilang saat singgah. Karena balasannya kumulatif (misalnya, $RR6$ berarti semua frame sampai 5 dibalas), kemungkinan A akan menerima RR urutannya sampai frame berikutnya dan akan tiba sebelum pewaktu yang dihubungkan dengan frame i berakhir.
- b. Bila pencatat waktu A habis, A mentransmisikan perintah RR sebagaimana dalam kasus 1b di atas. A menyusun pewaktu yang lain,

yang disebut pewaktu P-bit. Bila B gagal merespons perintah RR, atau bila responnya rusak, maka pewaktu P-bit A akan berakhir. Dalam hal ini, A akan kembali berusaha dengan cara membuat perintah R yang baru dan kembali mengulang pewaktu P-bit. Prosedur ini diusahakan untuk sejumlah iterasi. Bila A gagal memperoleh balasan setelah beberapa upaya maksimum dilakukan. A kembali mengulangi prosedur yang sama.

3. Rusaknya REJ. Bila REJ hilang, sama dengan kasus 1b.

Untuk bidang bernomorurut k-bit, ukuran window maksimum dibatasi sampai $2^k - 1$. Diasumsikan nomorurut 3-bit (jarak urutan nomor = 8). Anggap sebuah station mengirim frame 0 dan menerima kembali RR 1 dan kemudian mengirim frame 1,2,3,4,5,6,7,0 dan menerima RR 1 yang lain. Ini berarti bahwa kedelapan frame sudah diterima dengan benar dan RR1 merupakan balasan kumulatif. Tetapi, juga bisa berarti bahwa kedelapan frame rusak atau hilang saat transit. Problem seperti ini dapat dihindari bila ukuran jendela maksimum dibatasi sampai 7 ($2^3 - 1$).

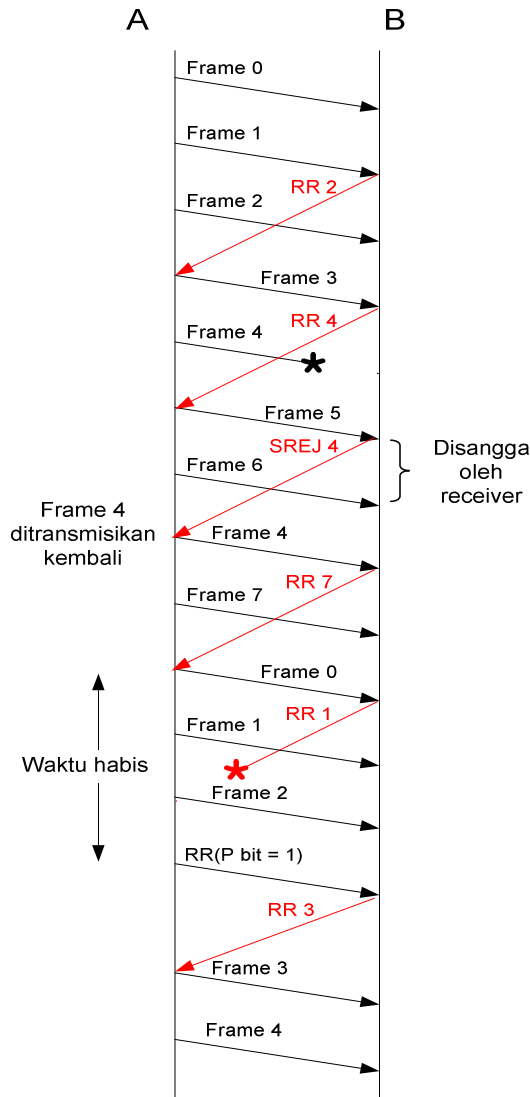
Gambar 2.7. Prosedur *Go-back-N* ARQ

2.6.3. *Selective-Reject* ARQ

Dengan *selective-reject* ARQ, frame-frame yang hanya ditransmisikan adalah frame-frame yang menerima balasan negatif, dalam hal ini disebut SREJ atau frame-frame yang waktunya sudah habis. Sebagai contoh, bila frame 5 diterima rusak, B mengirim SREJ 4, yang berarti frame 4 tidak diterima. Selanjutnya, B berlanjut dengan menerima frame-frame yang datang dan menahan

mereka sampai frame 4 yang *valid* diterima. Dalam hal ini, B dapat meletakkan frame sesuai pada tempatnya agar bisa dikirim ke *software* pada lapisan yang lebih tinggi.

Selective-reject lebih efisien dibanding *go-back-N*, karena *selective-reject* meminimalkan jumlah retransmisi. Dengan kata lain, *receiver* harus mempertahankan penyangga sebesar mungkin untuk menyimpan tempat bagi frame SREJ sampai frame yang rusak diretransmisi, serta harus memuat logika untuk diselipkan kembali frame tersebut pada urutan yang tepat. Selain itu, *transmitter* juga memerlukan logika yang lebih kompleks agar mampu mengirimkan frame di luar urutan. Karena komplikasi seperti itu, *select-reject* ARQ tidak terlalu banyak dipergunakan dibanding *go-back-N* ARQ.



Gambar 2.8. Prosedur *Selective-Reject* ARQ

2.7. Clock Tick

Clock tick sering juga disebut dengan *tick* atau *cycle*. *Tick* merupakan unit waktu terkecil yang dikenal oleh komputer. Semakin cepat *clock tick* atau *cycle*, maka semakin banyak instruksi yang dapat dijalankan CPU dalam satu detik. Kecepatan *clock tick* diekspresikan dalam megahertz atau gigahertz. Setiap instruksi yang akan dijalankan oleh komputer memerlukan sejumlah *clock tick*, tetapi adakalanya beberapa instruksi dapat dieksekusi dalam satu *clock tick* pada

komputer yang cepat. Di dalam program animasi, *tick* merupakan satuan waktu yang digunakan sebagai iterasi antar proses atau mekanisme animasi pada komputer.

2.8. Animasi

Animasi adalah rangkaian gambar yang membentuk sebuah gerakan. Salah satu keunggulan animasi dibanding dengan media lain seperti teks adalah kemampuannya untuk menjelaskan perubahan keadaan tiap waktu. Hal ini sangat membantu dalam menjelaskan prosedur dan urutan kejadian.

Terdapat 3 jenis format animasi, yaitu :

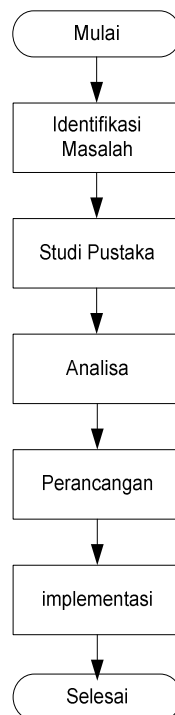
1. Animasi tanpa sistem kontrol, animasi ini hanya memberikan gambaran kejadian sebenarnya (*behavioural realism*), tanpa ada kontrol sistem. Misalnya untuk pause, memperlambat kecepatan penggantian frame, zoom in, zoom out, dan lain-lain.
2. Animasi dengan sistem kontrol, animasi ini dilengkapi dengan tombol kontrol misalnya tombol untuk pause, zoom in, zoom out, dll. Animasi dengan sistem control memungkinkan pengguna untuk menyesuaikan animasi dengan kapasitas pemrosesan informasi.
3. Animasi manipulasi langsung. (*Direct-Manipulation Animation (DMA)*). DMA menyediakan fasilitas untuk pengguna berinteraksi langsung dengan control navigasi (misal tombol dan slider). Pengguna bebas untuk menentukan arah perhatian. Menekan tombol atau menggeser slider akan menyebabkan perubahan keadaan. Hasilnya dapat langsung dilihat dan kejadiannya dapat diulang-ulang.

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan dipaparkan tahapan-tahapan yang dilakukan dalam penelitian agar hasil yang dicapai tidak menyimpang dari tujuan yang telah dilakukan sebelumnya.

Metodologi penelitian yang digunakan dalam penyusunan tugas akhir ini akan melalui beberapa tahapan yang membentuk sebuah alur yang sistematis. Tahap-tahap yang akan dilalui adalah sebagai berikut :



Gambar 3.1. Tahapan Pelaksanaan Tugas Akhir

3.1. Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan sebelumnya pada bab I, maka permasalahan yang diangkat pada tugas akhir ini adalah bagaimana membuat suatu animasi pembelajaran algoritma ARQ (*Automatic Repeat Request*) pada pengontrolan kesalahan pengiriman data dalam bentuk frame.

3.2. Studi Pustaka (*Library Research*)

Studi kepustakaan atau kajian pustaka dilakukan untuk mencari dan mempelajari serta mendalami informasi tentang pengontrolan kesalahan dan metode ARQ. Sumber kepustakaan diambil karya ilmiah yang berasal dari buku-buku maupun internet. Karya ilmiah yang dimaksud adalah berupa tulisan ilmiah yang berbentuk artikel, prosiding, buku, *e-book* (buku elektronik), dan lainnya. Studi pustaka bertujuan untuk mengetahui metode apa yang akan digunakan untuk menyelesaikan permasalahan yang akan diteliti, dan menerapkan suatu metode yang akan digunakan dalam tugas akhir ini.

3.3. Analisa

Setelah melakukan kajian pustaka, langkah berikutnya adalah menganalisa hasil dari teori pendukung tersebut. Mempelajari prosedur pengontrolan kesalahan pada algoritma *stop-and-wait* ARQ, *Go-back-N* ARQ, dan *Selective-reject* ARQ. Menganalisa kebutuhan-kebutuhan animasi dan merancang suatu model pembelajaran animasi. Hasil analisa kemudian dilanjutkan dengan perancangan antar muka.

3.4. Perancangan

Dari hasil analisa, dibuatlah suatu perancangan dari animasi pembelajaran ini menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0* dengan rancangan sebagai berikut:

- a. Perancangan *interface* animasi.
- b. Perancangan parameter inputan animasi.

Setiap perancangan *interface* dari ketiga metode ARQ, diberi parameter *input-an* berupa:

1. Waktu transmisi per frame.
2. Interval waktu *time out*.
3. Waktu transmisi per balasan.
4. Banyak frame yang ditransmisikan.
5. Kecepatan animasi.
6. Balasan RR (*receive ready*)
7. Untuk metode *go-back-n* ARQ dan *selective-reject* ARQ, yang menggunakan *sliding window flow control*, inputannya berupa urutan nomor bit dan ukuran jendela.

3.5. Implementasi

Hasil rancangan yang telah dilakukan pada tahap desain kemudian diubah menjadi bentuk yang dimengerti oleh mesin menggunakan bahasa pemrograman agar bisa diimplementasikan oleh pengguna. Tahapan ini mencakup penulisan kode program. Penulisan kode program dilakukan dengan menggunakan *Microsoft Virtual Basic 6.0* .

BAB IV

ANALISIS DAN PERANCANGAN

4.1. Analisis Animasi

Analisis dibutuhkan sebagai bahan acuan dalam membuat perangkat lunak animasi. Hasil animasi tersebut dapat dikembangkan ke dalam sebuah bentuk yang dapat diimplementasikan strukturnya dan dapat didefinisikan secara rinci.

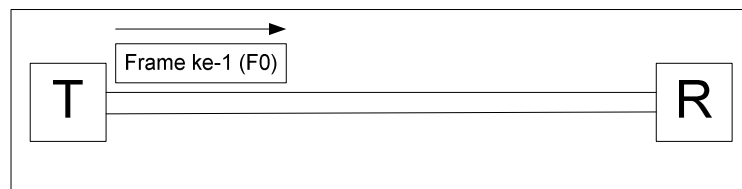
4.1.1. Analisis Animasi Metode *Stop-and-Wait* ARQ

Pada bagian ini akan dibangun suatu kondisi yang dapat menampilkan kondisi *stop-and-wait* dengan menggunakan animasi. Animasi dimodelkankan pada sebuah pengiriman dan penerimaan frame, dimana alur kerjanya *Transmitter* harus menunggu balasan dari *receiver* setelah mengirimkan frame. Tidak ada frame yang akan dikirimkan sampai balasan (ACK) dari *receiver* tiba di station sumber. *Transmitter* memberi penomoran 0 dan 1 bergantian pada frame-frame yang akan dikirim. *Receiver* menandai balasan (ACK) dengan ACK0 dan ACK1. ACK0 menyatakan *receiver* siap untuk menerima frame bernomor 0 sedangkan ACK1 menyatakan *receiver* siap untuk menerima frame bernomor 1. Sebagai contoh, *transmitter* mengirimkan frame 0. *Receiver* menerima frame 0 dengan baik dan harus membalas dengan ACK1. *Transmitter* menerima ACK1 dan melanjutkan pengiriman dengan mengirimkan frame 1. Selanjutnya, *receiver* menerima frame 1 dengan baik dan membalas dengan ACK0. Demikian pengiriman frame ini berjalan terus. Dengan penomoran seperti ini, *receiver* dapat mengontrol penerimaan frame secara berurutan.

Agar lebih jelas dalam memahami metode ini, perhatikanlah kasus berikut.

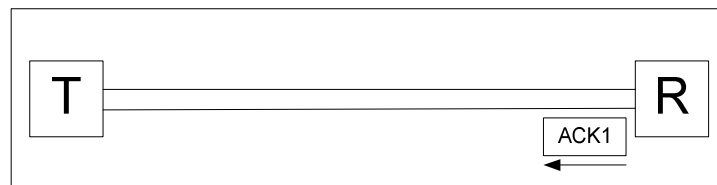
a. Proses pengiriman frame yang normal (tanpa gangguan).

1. *Transmitter* mengirimkan frame ke-1 bernomor 0 (F0).



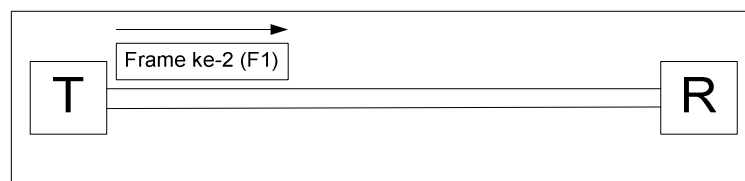
Gambar 4.1. *Transmitter* mengirimkan frame ke-1 (F0)

2. *Receiver* menerima frame ke-1 (F0) dengan baik dan membalas dengan mengirimkan ACK1 yang menyatakan *receiver* siap menerima frame bernomor 1 (F1).



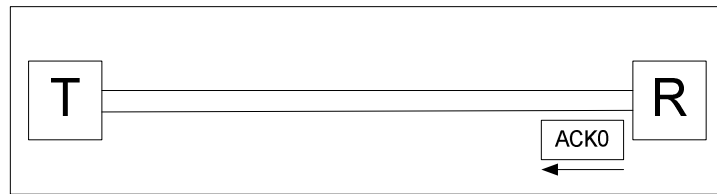
Gambar 4.2. *Receiver* mengirimkan ACK1

3. *Transmitter* menerima ACK1 dengan baik dan mengirimkan frame selanjutnya, frame ke-2 bernomor 1 (F1).



Gambar 4.3. *Transmitter* mengirimkan frame ke-2 (F1)

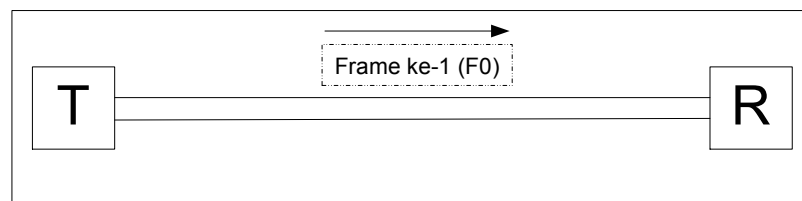
4. *Receiver* menerima frame ke-2 (F1) dengan baik dan membalas dengan mengirimkan ACK0 yang menyatakan *receiver* siap menerima F0 (frame berikutnya).



Gambar 4.4. *Receiver* mengirimkan ACK0

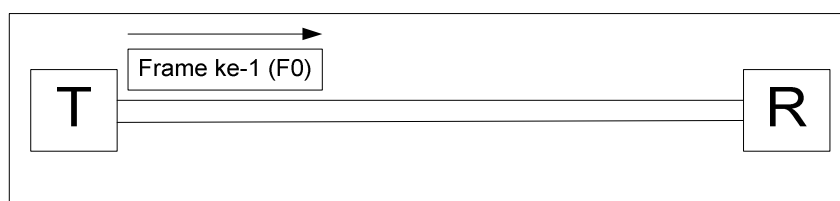
b. Proses pengiriman frame dengan gangguan berupa frame hilang atau frame rusak.

1. *Transmitter* mengirimkan frame ke-1 bernomor 0 (F0). Frame hilang di tengah proses transmisi.



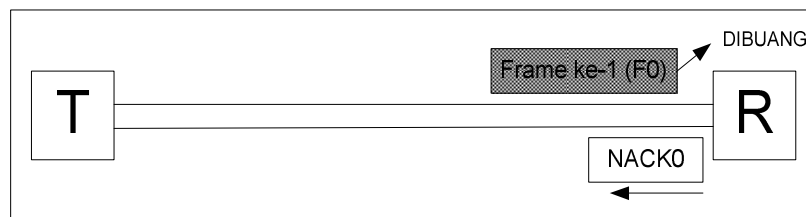
Gambar 4.5. *Transmitter* mengirimkan frame ke-1 (F0) dan frame hilang di tengah proses transmisi

2. *Receiver* tidak mengirimkan ACK. Pencatat waktu pada *transmitter* habis (*time out*). *Transmitter* kembali mengirimkan frame yang sama.



Gambar 4.6. *Transmitter* kembali mengirimkan frame ke-1 (F0)

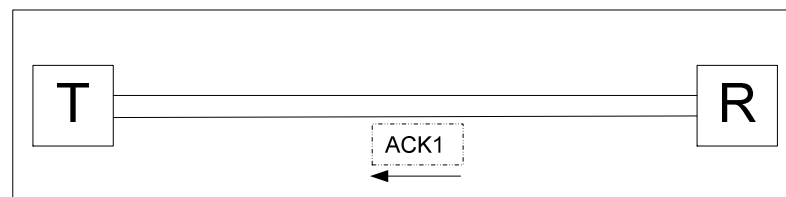
3. Untuk kasus frame mengalami kerusakan di tengah proses transmisi. *Receiver* menganggap frame tidak *valid*, sehingga membuang frame dan mengirimkan NACK terhadap nomor frame yang rusak. *Transmitter* menerima NACK dan kembali mengirimkan frame yang sama.



Gambar 4.7. Frame rusak dibuang dan *receiver* mentransmisikan NACK0

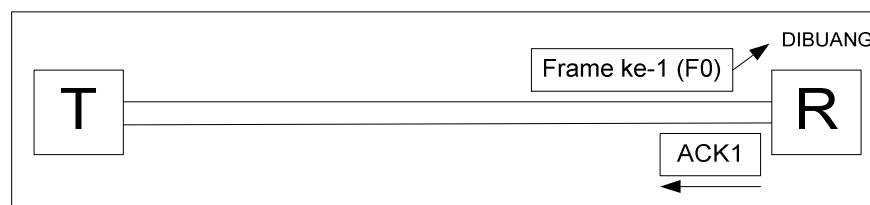
c. Proses pengiriman frame dengan gangguan berupa ACK hilang atau ACK rusak.

1. *Transmitter* mengirimkan frame ke-1 bernomor 0 (F0).
2. *Receiver* menerima frame ke-1 (F0) dengan baik dan membalas dengan mengirimkan ACK1. ACK1 hilang di tengah proses transmisi.



Gambar 4.8. ACK1 hilang di tengah proses transmisi

3. Pencatat waktu pada *transmitter* habis (*time out*). *Transmitter* kembali mengirimkan frame ke-1 bernomor 0 (F0).
4. Frame ke-1 (F0) tiba dengan baik di *receiver*. Karena *receiver* sedang mengharapkan frame bernomor 1 (F1), maka frame ini dibuang dan *receiver* kembali mentransmisikan ACK1.

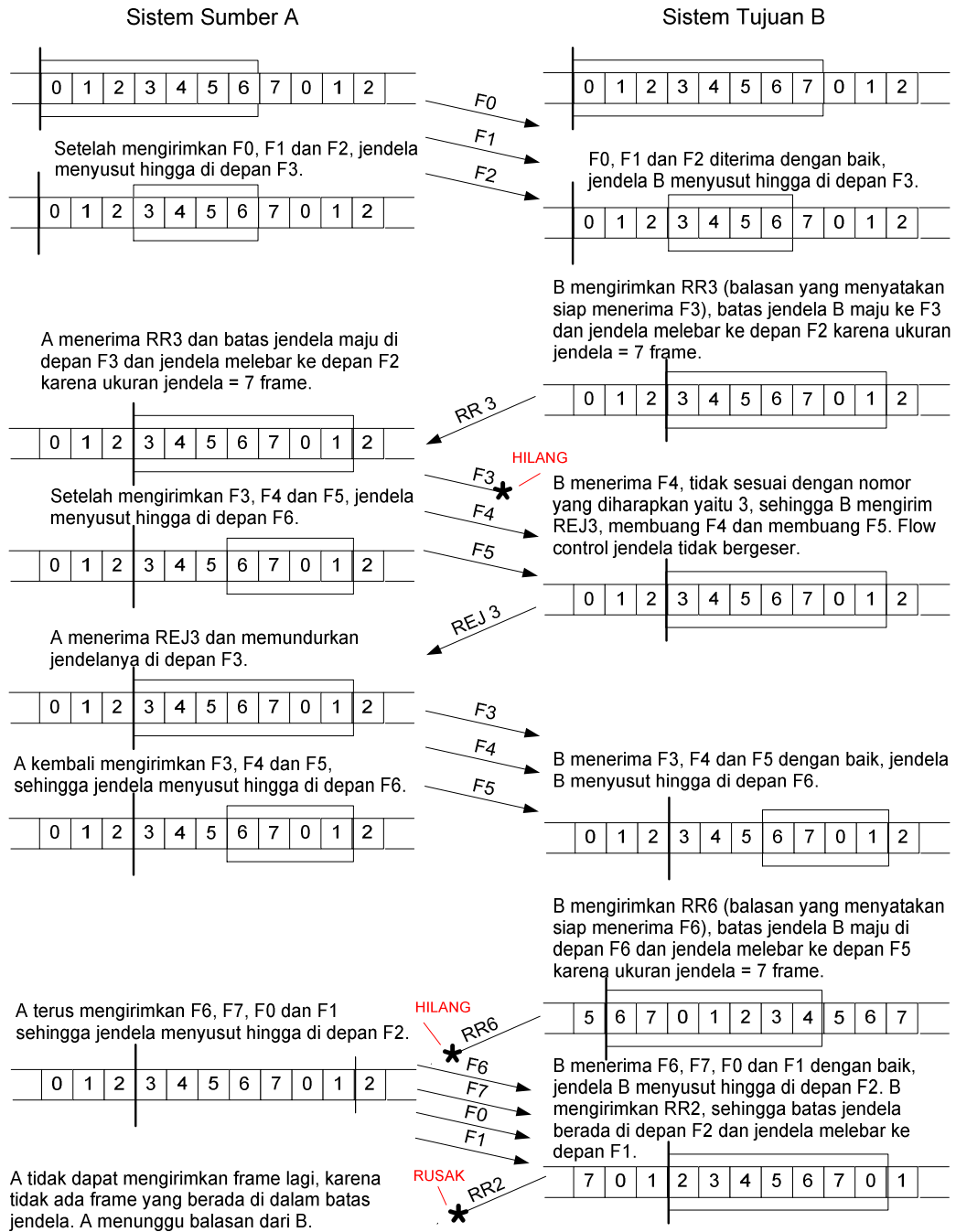


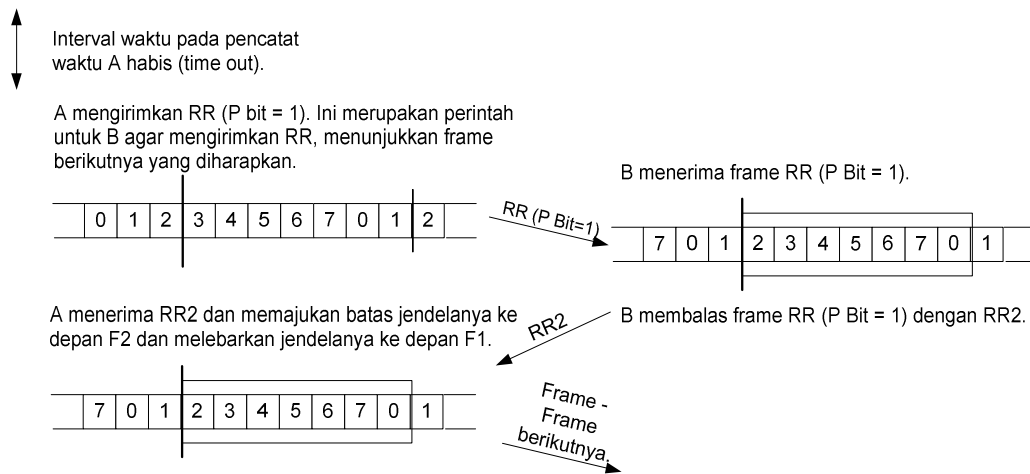
Gambar 4.9. Frame dibuang dan *receiver* mentransmisikan ACK1

5. Untuk kasus ACK / NACK rusak atau NACK hilang, sama dengan kasus ACK hilang.

4.1.2. Analisis Animasi Metode *Go-back-N* ARQ

Pada bagian ini akan dibangun kondisi deretan pengiriman frame. Station sumber bisa mengirim deretan frame yang diurutkan berdasarkan suatu modul bilangan. Metode ini menggunakan teknik *sliding window flow control* (kontrol arus jendela penggeseran). Bila tidak terjadi suatu kesalahan, station tujuan akan membalas (RR = *Receive Ready*) frame yang datang seperti biasa. Bila station tujuan mendeteksi suatu kesalahan pada sebuah frame, station tujuan mengirim balasan negatif (REJ = *Reject*) untuk frame tersebut. Station tujuan kemudian membuang frame itu dan semua frame-frame yang nantinya akan datang sampai frame yang mengalami kesalahan diterima dengan benar. Jadi, station sumber jika menerima REJ, harus melakukan retransmisi terhadap frame yang mengalami kesalahan tersebut plus semua frame pengganti yang ditransmisikan sementara. Agar lebih jelas dalam memahami metode ini, perhatikanlah contoh berikut. Misalkan, ukuran jendela = 7 frame dan penomoran frame dimulai dari 0 sampai 7.





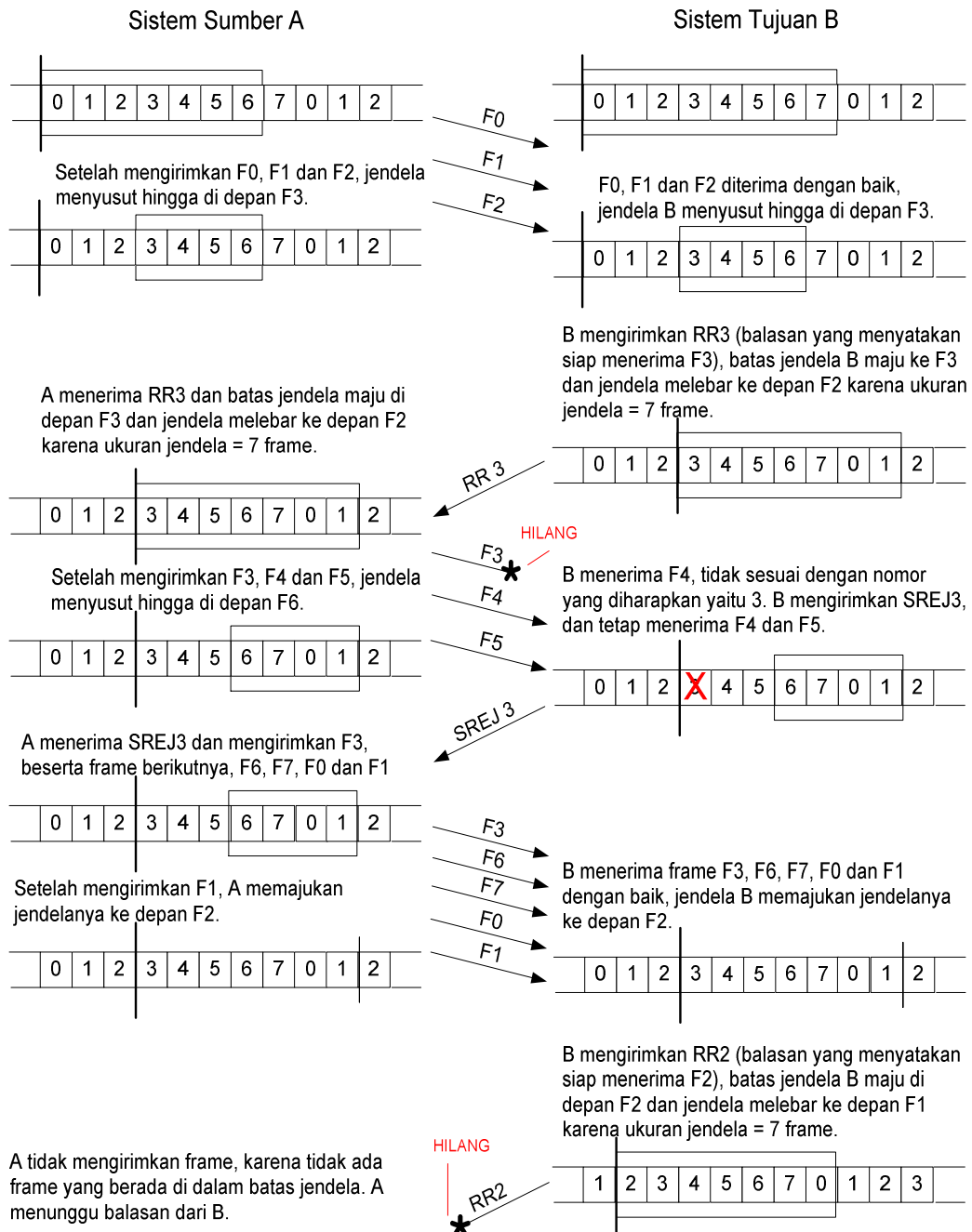
Gambar 4.10. Contoh kasus animasi metode *go-back-n* ARQ

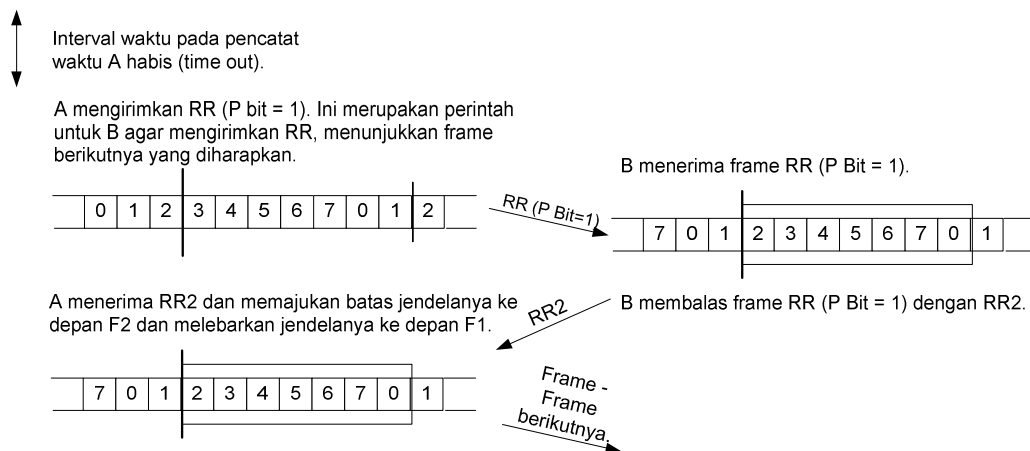
Dari contoh kasus di atas, dapat terlihat bahwa aliran frame untuk metode ini bersifat kontinyu (terus menerus) hingga frame yang berada di dalam jendela penggeseran habis. Balasan kumulatif oleh *receiver* akan memajukan dan melebarkan jendela penggeseran *transmitter* dan membuat *transmitter* kembali mengirimkan frame kembali. Dalam metode ini, untuk penomoran bernomorurut k -bit, yang menyediakan jarak bernomorurut 2^k , ukuran jendela maksimum dibatasi sampai $2^k - 1$. Misalnya, untuk penomoran 3-bit, jarak nomorurut = $2^3 = 8$ (frame diberi penomoran 0 sampai 7), ukuran jendela maksimum adalah $2^3 - 1 = 8 - 1 = 7$ frame.

4.1.3. Analisis Animasi Metode *Selective-Reject* ARQ

Mekanisme tampilan proses kerja *selective-reject* ini hampir sama dengan animasi *Go-back-N* ARQ. Perbedaannya adalah dalam metode ini, frame-frame yang diretransmisikan hanya frame-frame yang menerima balasan negatif (SREJ). Bila frame 4 diterima rusak, maka *receiver* akan mengirim SREJ 4, yang berarti frame 4 tidak diterima. Selanjutnya, *receiver* berlanjut dengan menerima frame-

frame yang datang dan menahan mereka sampai frame 4 yang *valid* diterima. Agar lebih jelas, perhatikanlah contoh berikut. Misalkan, ukuran jendela = 7 frame dan penomoran frame dimulai dari 0 sampai 7.





Gambar 4.11. Contoh kasus animasi metode *selective-reject* ARQ

Metode ini lebih efisien dibandingkan dengan metode *go-back-n* ARQ, karena metode ini meminimalkan jumlah retransmisi. Kekurangan dari metode ini adalah *transmitter* dan *receiver* memerlukan logika yang lebih kompleks agar mampu mengirimkan frame di luar urutan dan menyelipkan kembali frame pada urutan yang tepat. Karena komplikasi semacam itu, *Selective-Reject* ARQ tidak terlalu banyak dipergunakan dibanding *Go-back-N* ARQ.

4.1.4. Analisis Data Masukan Menggunakan Metode *Stop-and-Wait* ARQ

Parameter masukan menggunakan metode *stop-and-wait* ARQ terdiri dari 3 (tiga) bagian, diantaranya :

1. *Transmitter* (sumber), terdiri dari :
 - a. Waktu transmisi per frame.
 - b. Interval waktu *time out*.
 - c. Banyak frame yang ditransmisikan.
2. *Receiver* (tujuan), berisikan waktu transmisi per balasan.
3. Kecepatan animasi.

4.1.5. Analisis Data Masukan Menggunakan Metode *Go-Back-N* ARQ

Parameter masukan menggunakan metode *go-back-n* ARQ terdiri dari 4 (empat) bagian, diantaranya :

1. *Transmitter* (sumber), terdiri dari:
 - a. Waktu transmisi per frame.
 - b. Interval waktu *time out*
 - c. Banyak frame yang ditransmisikan.
2. *Receiver* (tujuan), terdiri dari:
 - a. Waktu transmisi per balasan.
 - b. Balasan RR (*Receive Ready*)
3. *Sliding window flow control*, terdiri dari:
 - a. Urutan nomor frame.
 - b. Ukuran window (jendela).
4. Kecepatan animasi.

4.1.6. Analisis Data Masukan Menggunakan Metode *Selective-Reject* ARQ

Parameter masukan menggunakan metode *selective-reject* ARQ terdiri dari 4 (empat) bagian, diantaranya:

1. *Transmitter* (sumber), terdiri dari:
 - a. Waktu transmisi per frame.
 - b. Interval waktu *time out*
 - c. Banyak frame yang ditransmisikan.
2. *Receiver* (tujuan), terdiri dari:
 - a. Waktu transmisi per balasan.

- b. Balasan RR (*Receive Ready*)
- 3. *Sliding window flow control*, terdiri dari:
 - a. Urutan nomor frame.
 - b. Ukuran window (jendela).
- 4. Kecepatan animasi.

4.1.7. Analisis Proses Pengaturan Waktu Animasi

Proses dalam pembuatan animasi pembelajaran ARQ (*Automatic Repeat Request*) pada pengontrolan kesalahan akan dianalisa prosedur kerjanya dalam 3 (tiga) metode pengontrolan kesalahan, yaitu metode *Stop-and-Wait* ARQ, metode *Go-back-N* ARQ dan *Selective-Reject* ARQ.

1. Proses Pengaturan Waktu Animasi
2. Proses Transmisi Frame oleh *Transmitter*.
3. Proses Pengiriman Frame.
4. Proses Penerimaan Frame oleh *Receiver*.
5. Proses Transmisi Balasan oleh *Receiver*.
6. Proses Pengiriman Balasan.
7. Proses Penerimaan Balasan oleh *Transmitter*.

Pada proses pengaturan waktu animasi berfungsi untuk mengatur penambahan waktu setiap *tick*-nya di dalam program animasi sesuai dengan skala waktu yang di-*input*, menjalankan pencatat waktu, dan prosedur yang harus dilakukan apabila terjadi '*time-out*'.

a. Proses pengaturan waktu animasi (metode *stop-and-wait* ARQ)

<Prosedur di dalam timer 'tmrWaktu'>

1. Tambah Waktu dengan 1 *tick*
2. Jika Status <> "TRANSMITTING FRAME" Maka Jalankan pencatat waktu dengan mengurangi interval waktu 'time-out' dengan 1 *tick*
3. Jika waktu pada pencatat waktu habis atau terjadi 'time-out' Maka Reset waktu pada pencatat waktu menjadi penuh kembali 'Mulai waktu transmit dan 'Ubah status menjadi transmitting frame

<End Prosedur>

b. Proses pengaturan waktu animasi (metode *go-back-n* dan *selective-reject* ARQ)

<Prosedur di dalam timer 'tmrWaktu'>

1. Tambah Waktu dengan 1 *tick*
2. Jika Transmitter tidak sedang mentransmisikan frame, Maka Jalankan pencatat waktu dengan mengurangi interval waktu 'time-out' dengan 1 *tick*.
3. Jika waktu pada pencatat waktu habis atau terjadi 'time-out' Maka Reset waktu pada pencatat waktu menjadi penuh kembali 'Perintahkan Transmitter untuk mengirim RR (P Bit = 1)

<End Prosedur>

4.1.8. Proses Transmisi Frame oleh *Transmitter*

Berfungsi untuk mengatur proses transmisi frame oleh *transmitter*. (Di dalam tampilan animasi, proses transmisi frame digambarkan dengan berkedip-kedipnya layar monitor *transmitter*).

a. Proses transmisi frame oleh *transmitter* (metode *stop-and wait* ARQ)

<Prosedur di dalam timer 'tmrProses'>

1. Memulai proses pengiriman frame dengan membuat layar monitor transmitter berkedip-kedip.
2. Jika Waktu Transmit PerFrame + Waktu Mulai Transmit <= Waktu sekarang, maka Reset waktu pada pencatat waktu menjadi penuh kembali
3. Munculkan objek frame yang siap untuk dikirimkan dan Beri label pada objek frame

<End Prosedur>

b. Proses transmisi frame oleh *transmitter* (metode *go-back-n* dan *selective-reject ARQ*)

<Prosedur di dalam timer 'tmrTransmitter'>

1. Jika ada frame yang akan dikirim Maka Memulai proses pengiriman frame dengan membuat layar monitor transmitter berkedip-kedip.
2. Jika Waktu Transmit PerFrame + Waktu Mulai Transmit \leq Waktu sekarang, maka Reset waktu pada pencatat waktu menjadi penuh kembali.
3. Maju dan Mundurkan jendela dengan teknik sliding window flow control.
4. Jika Transmitter mengirim frame = "RR P Bit = 1" maka Bentuk sebuah objek frame baru.

<End Prosedur>

4.1.9. Analisis Proses Pengiriman Frame

Berfungsi untuk mengatur proses perambatan frame pada medium dan dapat diubah level kecepatannya. Dalam proses ini, terdapat 2 gangguan yaitu frame rusak atau frame hilang. Secara teknis, proses ini berfungsi untuk memajukan frame-frame (ke arah kanan) dan memvisualisasikan gangguan yang dipilih *user* pada frame yang sedang dikirim.

a. Proses pengiriman frame (metode *stop-and-wait ARQ*)

<Prosedur di dalam timer 'tmrProses'>

1. 'Geser frame ke kanan sebanyak nSpeed (nSpeed sesuai dengan level 'kecepatan yang dipilih)
2. Jika frame sudah sampai ke Receiver Maka 'Ubah status Status = "RECEIVING FRAME"
Dengan mengubah status menjadi "RECEIVING FRAME", maka untuk eksekusi berikutnya, timer akan menjalankan algoritma penerimaan frame oleh receiver)
3. Jika frame sedang berada di tengah perjalanan dan diberi gangguan rusak Maka 'VISUALISASI FRAME RUSAK DENGAN MEMBERI TANDA SILANG 'PADA FRAME
4. Jika tombol 'Frame Hilang' dipilih Maka 'HILANGKAN FRAME

<End Prosedur>

b. Proses pengiriman frame (metode *go-back-n* dan *selective-reject ARQ*)

<Prosedur di dalam timer 'tmrMedium'>

1. Geser frame ke kanan sebanyak nSpeed (nSpeed sesuai 'dengan level kecepatan yang dipilih)

2. Jika frame telah sampai ke Receiver Maka <Jalankan algoritma penerimaan frame oleh Receiver> 'Hilangkan frame yang telah diterima.
 3. Jika frame sedang berada di tengah perjalanan dan sampai pada receiver maka frame berada dalam keadaan baik
 4. Jika tombol 'Frame Rusak' dipilih Maka 'VISUALISASI FRAME RUSAK DENGAN MEMBERI TANDA 'SILANG PADA FRAME.
 5. Jika tombol frame hilang dipilih Maka 'HILANGKAN FRAME.
- <End Prosedur>

4.1.10. Analisis Proses Penerimaan Frame oleh Receiver

Berfungsi untuk mengatur proses penerimaan frame oleh receiver.

Proses yang dibahas ini adalah respons dari receiver setelah menerima frame dalam keadaan baik atau tidak baik.

a. Proses penerimaan frame oleh receiver (metode *stop-and-wait* ARQ)

<Prosedur di dalam timer 'tmrProses'>
 'FRAME SAMPAI KE RECEIVER
 1. Jika frame tidak bertanda silang Maka 'FRAME DALAM KEADAAN BAIK
 2. Jika IDACK = IDFrame Maka 'FRAME SESUAI dan Jika tidak maka 'FRAME TIDAK SESUAI dan 'Gambar komputer receiver - blink Warna Merah
 3. Eksekusi berikutnya adalah proses transmisi balasan oleh Receiver.
 Status = "TRANSMITTING ACK"
 <End Prosedur>

b. Proses penerimaan frame oleh receiver (metode *go-back-n* ARQ)

<Prosedur di dalam timer 'tmrMedium'>
 'FRAME SAMPAI KE RECEIVER
 1. Jika frame tidak bertanda silang Maka FRAME DALAM KEADAAN BAIK.
 2. Jika label frame yang diterima sesuai dengan label window dari flow control receiver Maka 'FRAME SESUAI 'Majukan Flow Control Jendela Pergeseran Receiver
 3. Panggil Prosedur Gambar Flow Control Receiver
 4. Jika label frame yang diterima tidak sesuai maka receiver mengirim REJ 'jendela flow control tetap.
 <End Prosedur>

c. Proses penerimaan frame oleh receiver (metode *selective-reject* ARQ)

<Prosedur di dalam timer 'tmrMedium'>
 'FRAME SAMPAI KE RECEIVER
 1. Jika frame tidak bertanda silang Maka FRAME DALAM KEADAAN BAIK.

2. Jika label frame yang diterima sesuai dengan label window dari flow control receiver Maka 'FRAME SESUAI 'Majukan Flow Control Jendela Pergeseran Receiver
3. Panggil Prosedur Gambar Flow Control Receiver
4. Jika label frame yang diterima tidak sesuai dengan label window maka receiver mengirim SREJ 'jendela flow control ditandai frame silang.

<End Prosedur>

4.1.11. Analisis Proses Transmisi Balasan oleh *Receiver*

Berfungsi untuk mengatur proses Transmisi balasan oleh *receiver*. (Di dalam animasi ini, proses transmisi balasan digambarkan dengan berkedip-kedipnya layar monitor *receiver*).

a. Proses transmisi balasan oleh *receiver* (metode *stop-and-wait* ARQ)

<Prosedur di dalam timer 'tmrProses'>

1. Memulai proses pengiriman balasan dengan membuat layar monitor receiver berkedip-kedip. 'Proses transmisi selesai
2. Jika Waktu Transmit PerBalasan + Waktu Mulai Transmit \leq Waktu sekarang, maka Munculkan objek balasan ACK yang siap untuk dikirimkan' Beri label pada objek balasan.
3. Set interval timer - proses. 'Ubah status menjadi = "SENDING ACK".

<End Prosedur>

b. Proses transmisi balasan oleh *receiver* (metode *go-back-n* dan *selective-reject* ARQ)

<Prosedur di dalam timer 'tmrReceiver'>

1. Jika ada balasan yang akan dikirim Maka Memulai proses pengiriman balasan dengan membuat layar monitor receiver berkedip-kedip. 'Proses transmisi selesai
2. Jika Waktu Transmit PerBalasan + Waktu Mulai Transmit \leq Waktu sekarang, maka kirim frame balasan REJ atau SREJ
3. 'Majukan Flow Control dengan teknik sliding window flow control

<End Prosedur>

4.1.12. Analisis Proses Pengiriman Balasan

Berfungsi untuk mengatur proses perambatan balasan pada medium.

Dalam proses ini, terdapat 2 (dua) gangguan yaitu balasan rusak atau balasan

hilang. Secara teknis, berfungsi untuk memajukan frame balasan (ke arah kiri) dan memvisualisasikan gangguan yang dipilih *user* pada balasan yang sedang dikirim.

a. Proses pengiriman balasan (metode *stop-and-wait* ARQ)

```
<Prosedur di dalam timer 'tmrProses'>
  1. 'Geser balasan ke kiri sebanyak nSpeed (nSpeed sesuai dengan
     level 'kecepatan yang dipilih)
  2. Jika balasan sudah sampai ke Transmitter Maka 'Ubah status =
     "RECEIVING ACK".
     (Dengan mengubah status menjadi "RECEIVING ACK", maka untuk
     eksekusi berikutnya, timer akan menjalankan prose penerimaan
     balasan oleh transmitter)
  3. Jika balasan sedang berada di tengah perjalanan dan diterima
     receiver maka frame baik.
  4. Jika tombol 'ACK Rusak' dipilih Maka 'VISUALISASI ACK RUSAK
     DENGAN MEMBERI TANDA SILANG 'PADA ACK
  5. Jika tombol 'ACK Hilang' dipilih Maka HILANGKAN ACK.
<End Prosedur>
```

b. Proses pengiriman balasan (metode *go-back-n* dan *selective-reject* ARQ)

```
<Prosedur di dalam timer 'tmrMedium'>
  1. 'Geser balasan ke kiri sebanyak nSpeed (nSpeed sesuai
     'dengan level kecepatan yang dipilih)
  2. Jika balasan telah sampai ke Transmitter Maka <Jalankan
     proses penerimaan balasan oleh Transmitter>
     'Hilangkan balasan yang telah diterima
  3. Jika balasan sedang berada di tengah perjalanan dan sampai
     pada receiver maka balasan berada dalam keadaan baik.
  4. Jika tombol 'Balasan Rusak' dipilih Maka 'VISUALISASI
     BALASAN RUSAK DENGAN MEMBERI TANDA 'SILANG PADA BALASAN
  5. Jika tombol 'Balasan hilang' Maka HILANGKAN BALASAN
<End Prosedur>
```

4.1.13. Analisis Proses Penerimaan Balasan oleh *Transmitter*

Berfungsi untuk mengatur proses penerimaan balasan oleh *transmitter*.

Proses yang dibahas adalah respons dari *transmitter* setelah menerima balasan.

a. Proses penerimaan balasan oleh *transmitter* (metode *stop-and-wait* ARQ)

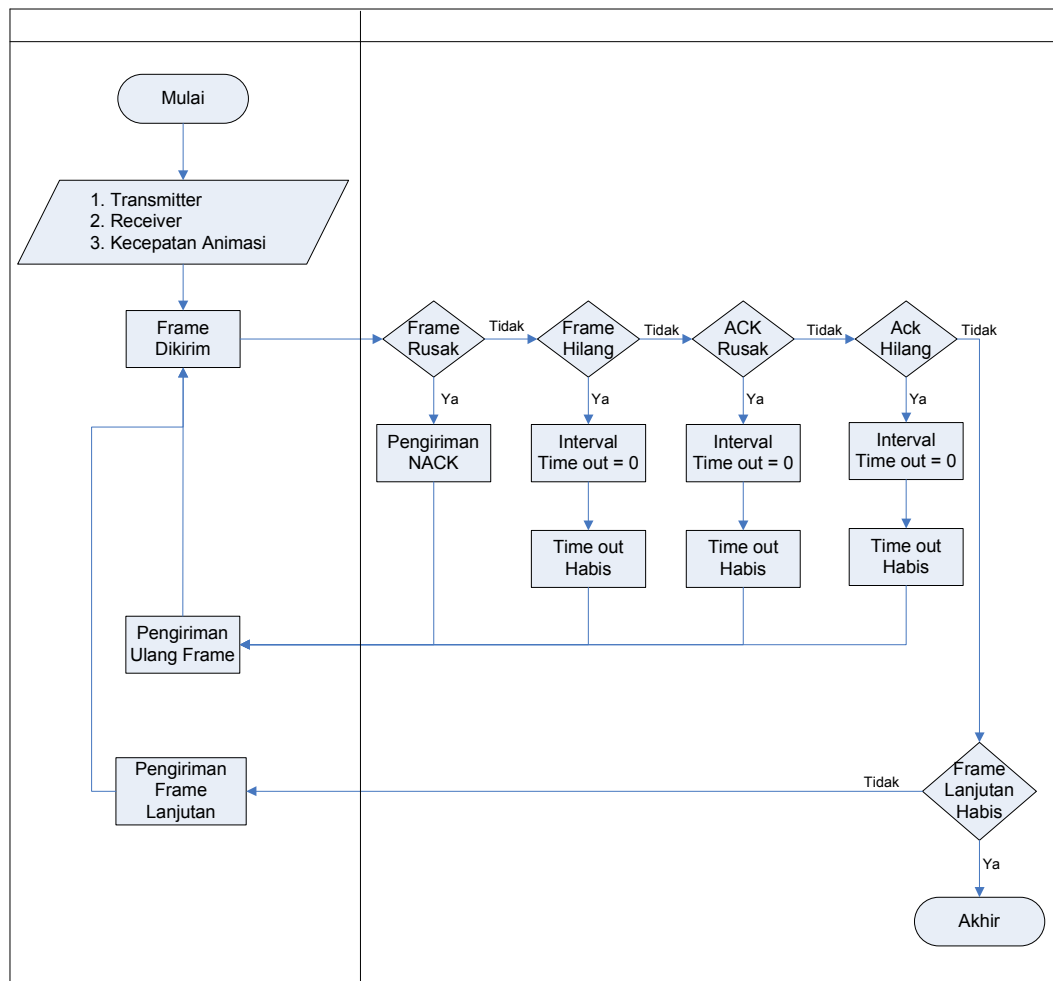
```
<Prosedur di dalam timer 'tmrProses'>
'BALASAN ACK SAMPAI KE TRANSMITTER
  1. Jika ACK tidak bertanda silang Maka 'ACK DALAM KEADAAN BAIK
     (Eksekusi berikutnya adalah proses transmisi frame oleh
     Transmitter) Status = "TRANSMITTING FRAME"
  2. Jika ACK bertanda silang, artinya ACK rusak.'Gambar komputer
     transmitter - blink Warna Merah. Maka Transmitter akan
     mengirim ulang frame
<End Prosedur>
```

b. Proses penerimaan balasan oleh *transmitter* (metode *go-back-n* ARQ)

```
<Prosedur di dalam timer 'tmrMedium'>
'BALASAN REJ SAMPAI KE TRANSMITTER
  1. Jika balasan tidak bertanda silang Maka 'REJ DALAM KEADAAN
    BAIK.
  2. Jika balasan yang diterima adalah RR Maka Reset waktu time-
    out pada pencatat waktu.
  3. Majukan Flow Control Jendela Pergeseran Transmitter. Panggil
    Prosedur Gambar Flow Control Transmitter
  4. Jika Transmitter menerima "REJ" Maka 'Mundurkan kembali flow
    control.Panggil Prosedur Gambar Flow Control Transmitter
<End Prosedur>
```

c. Proses penerimaan balasan oleh *transmitter* (metode *selective-reject* ARQ)

```
<Prosedur di dalam timer 'tmrMedium'>
'BALASAN SREJ SAMPAI KE TRANSMITTER
  1. Jika balasan tidak bertanda silang Maka 'SREJ DALAM KEADAAN
    BAIK
  2. Jika balasan yang diterima adalah RR Maka Reset waktu time-
    out pada pencatat waktu.
  3. Majukan Flow Control Jendela Pergeseran Transmitter. Panggil
    Prosedur Gambar Flow Control Transmitter
  4. Jika Transmitter menerima "SREJ" Maka tandai silang pada
    flow control.Panggil Prosedur Gambar Flow Control
    Transmitter
<End Prosedur>
```

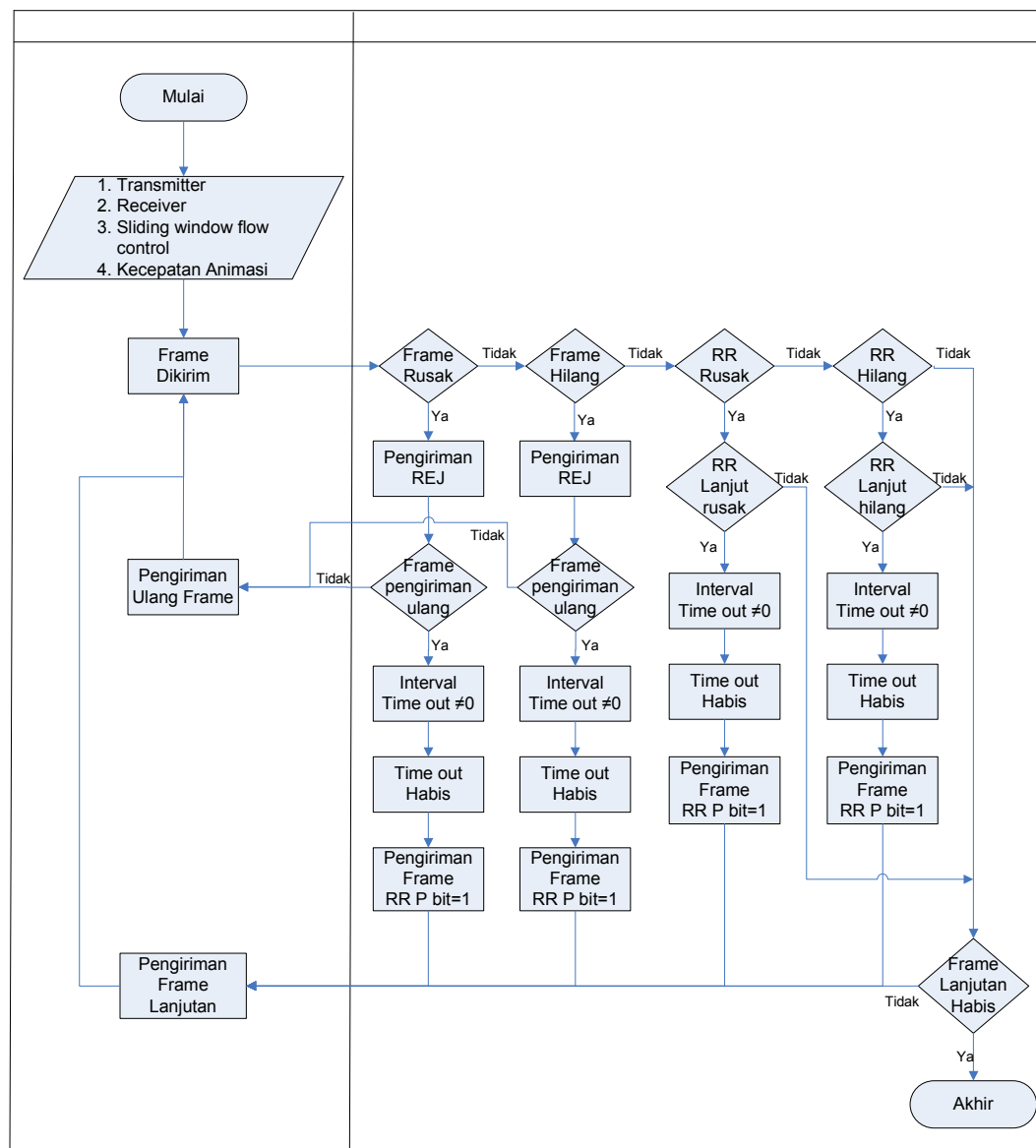


Gambar 4.12. Flowchart animasi algoritma *stop-and-wait* ARQ

Penjelasan dari gambar *flowchart* diatas adalah sebagai berikut:

- Input* nilai pada bagian *transmitter* yang berisi waktu transmisi per frame minimal 2 tick, interval time out minimal 13 tick, dan banyak frame yang ditransmisikan maksimal 50 buah.
- Input* nilai pada bagian *receiver* yang berisi waktu transmisi per balasan minimal 2 tick, nilai *input* kecepatan animasi minimal 100 tick dan maksimal 1000 tick.

- c. Selanjutnya memulai animasi dengan proses frame dikirim. Jika frame rusak, maka proses pengiriman NACK (*Negative Acknowledgment*) dan *transmitter* melakukan proses pengiriman ulang frame.
- d. Kemudian jika tidak pada kondisi frame rusak, maka memilih opsi kondisi frame hilang, ACK rusak, ACK hilang, dan jika kondisi frame lanjutan habis maka proses pengiriman frame selesai. Jika tidak maka proses pengiriman frame lanjutan.
- e. Kemudian jika frame hilang, maka proses perhitungan interval dan time out habis, *transmitter* akan melakukan pengiriman ulang frame. Jika tidak, memilih kondisi ACK rusak, ACK hilang, dan kondisi jika frame lanjutan habis maka proses pengiriman frame selesai. Jika tidak maka proses pengiriman frame lanjutan.
- f. Untuk kondisi balasan ACK rusak maupun ACK hilang pada proses pengiriman frame, sama dengan kondisi frame hilang.
- g. Selanjutnya tampilkan laporan keterangan kejadian (*history log*) dari proses animasi metode *stop-and-wait* ARQ.

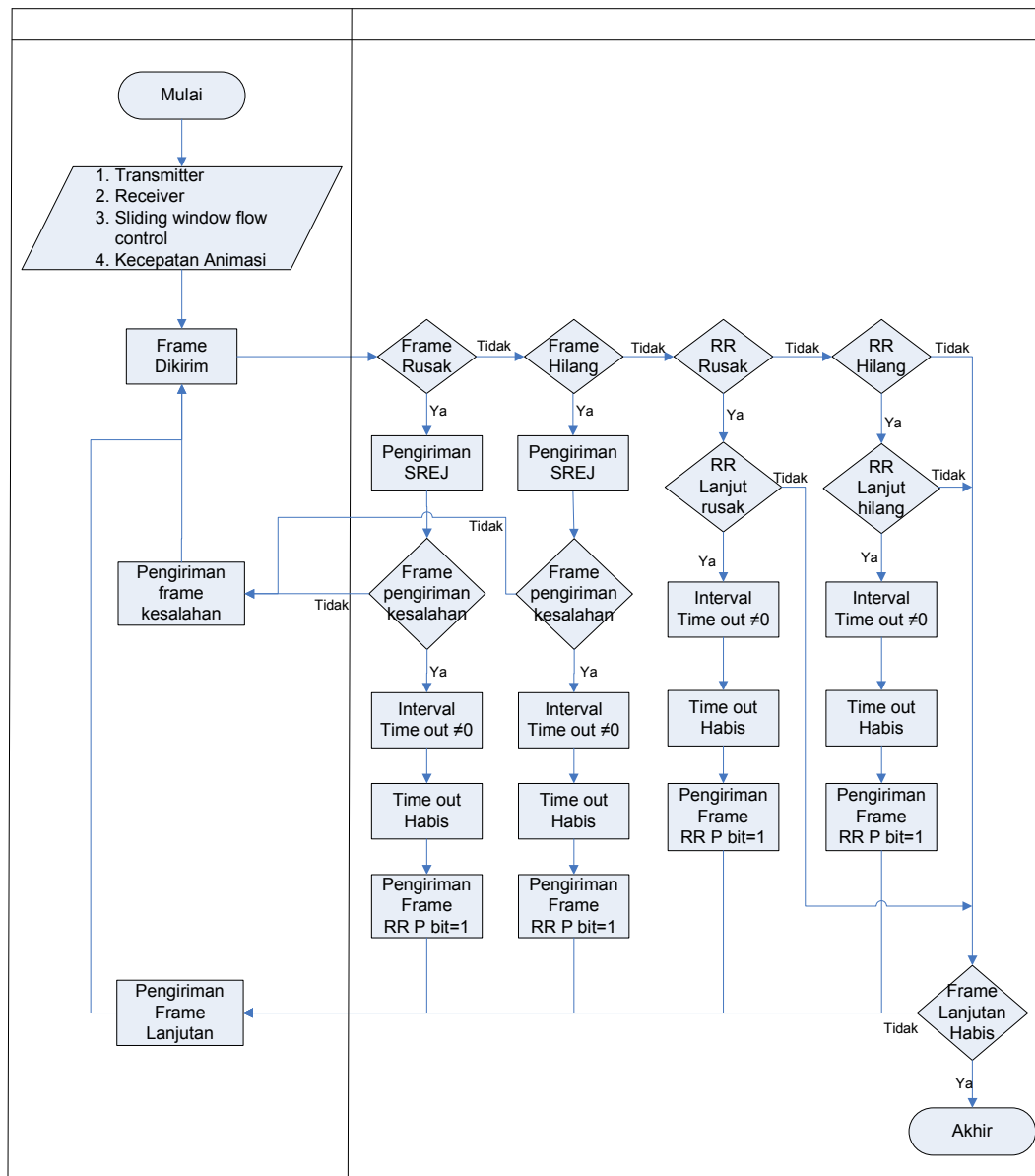


Gambar 4.13. Flowchart animasi algoritma go-back-n ARQ

Penjelasan dari gambar flowchart diatas adalah sebagai berikut:

- Input* nilai pada bagian transmitter yang berisikan waktu transmisi per frame minimal 2 tick, interval *time out* minimal 13 tick, dan banyak frame yang ditransmisikan maksimal 50 buah.

- b. *Input* nilai pada bagian *receiver* yang berisikan waktu transmisi per balasan minimal 2 tick, dan balasan RR (*Receive Ready*) dikirim setiap menerima *n*-frame dengan baik.
- c. *Input* sliding window flow control yang berisikan urutan nomor frame (2-5bit) dan ukuran jendela.
- d. Nilai *input* kecepatan animasi minimal 100 tick dan maksimal 1000 tick.
- e. Selanjutnya memulai animasi dengan proses frame dikirim. Jika frame rusak, maka proses pengiriman REJ (*Reject*) dan *transmitter* melakukan pengiriman ulang frame. Jika Ya pada kondisi frame pengiriman ulang, maka proses perhitungan interval dan *time out* habis, *transmitter* akan melakukan proses pengiriman frame RR $P_{bit} = 1$.
- f. Kemudian jika tidak pada kondisi frame rusak, maka memilih opsi kondisi frame hilang, RR rusak, RR hilang, dan jika kondisi frame lanjutan habis maka proses pengiriman frame selesai. Jika tidak maka proses pengiriman frame lanjutan. Untuk kondisi frame hilang sama dengan frame rusak.
- g. Kemudian jika RR rusak, maka berlanjut pada kondisi RR lanjut habis. Jika Ya maka proses perhitungan interval dan proses *time out* habis, maka *transmitter* akan melakukan proses pengiriman frame RR $P_{bit} = 1$. Jika tidak, dan kondisi frame lanjutan habis maka selesai. Jika tidak maka proses pengiriman frame lanjutan. Untuk kondisi RR hilang sama dengan RR rusak.
- h. Selanjutnya tampilkan laporan keterangan kejadian (*history log*) dari proses animasi metode *go-back-n* ARQ.



Gambar 4.14. Flowchart animasi algoritma selective-reject ARQ

Penjelasan dari gambar flowchart diatas adalah sebagai berikut:

- a. *Input* nilai pada bagian transmitter yang berisikan waktu transmisi per frame minimal 2 tick, interval *time out* minimal 13 tick, dan banyak frame yang ditransmisikan maksimal 50 buah.

- b. *Input* nilai pada bagian *receiver* yang berisikan waktu transmisi per balasan minimal 2 tick, dan balasan RR (*Receive Ready*) dikirim setiap menerima n-frame dengan baik.
- c. *Input* sliding window flow control yang berisikan urutan nomor frame (2-5bit) dan ukuran jendela.
- d. Nilai *input* kecepatan animasi minimal 100 tick dan maksimal 1000 tick.
- e. Selanjutnya memulai animasi dengan proses frame dikirim. Jika frame rusak, maka proses pengiriman SREJ dan *transmitter* hanya melakukan pengiriman frame kesalahan. Jika Ya pada kondisi frame pengiriman kesalahan, maka proses perhitungan interval dan proses *time out* habis, *transmitter* akan melakukan proses pengiriman frame RR Pbit =1.
- f. Kemudian jika tidak pada kondisi frame rusak, maka memilih opsi kondisi frame hilang, RR rusak, RR hilang, dan jika kondisi frame lanjutan habis maka proses pengiriman frame selesai. Jika tidak maka proses pengiriman frame lanjutan. Untuk kondisi frame hilang sama dengan frame rusak.
- g. Kemudian jika RR rusak, maka berlanjut pada kondisi RR lanjut habis. Jika Ya maka proses perhitungan interval dan proses *time out* habis, maka *transmitter* akan melakukan proses pengiriman frame RR Pbit = 1. Jika tidak, dan kondisi frame lanjutan habis maka selesai. Jika tidak maka proses pengiriman frame lanjutan. Untuk kondisi RR hilang sama dengan RR rusak
- h. Selanjutnya tampilkan laporan keterangan kejadian (*history log*) dari proses animasi metode *selective-reject* ARQ.

4.1.4. Analisis Data Keluaran (*output*).

Tujuan akhir dari pembuatan animasi pembelajaran algoritma ARQ pada pengontrolan kesalahan meliputi metode *Stop-and-Wait* ARQ, *Go-back-N* ARQ, *Selective-Reject* ARQ berupa keterangan proses (*history log*) pada pengiriman data dalam bentuk frame secara keseluruhan. Keterangan proses dari ketiga animasi ARQ ini menampilkan perbedaan waktu proses dalam mentransmisi seluruh frame.

4.2. Perancangan animasi

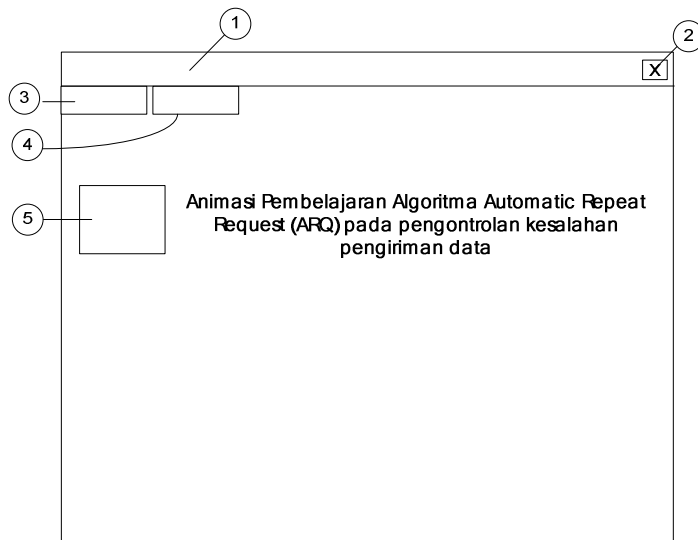
Animasi pembelajaran algoritma ARQ ini dirancang dengan menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0*. dan dirancang dengan menggunakan beberapa komponen standard seperti *command button*, *text box*, *up down*, *option button*, *label*, *image*, *picture box*, *shape*, dan sebagainya.

Animasi pembelajaran ini memiliki beberapa *form*, antara lain :

- a. *Form* Utama.
- b. *Form Input* Animasi Metode *Stop-and-Wait* ARQ.
- c. *Form* Animasi Metode *Stop-and-Wait* ARQ.
- d. *Form Input* Animasi Metode *Go-back-N* ARQ.
- e. *Form* Animasi Metode *Go-back-N* ARQ.
- f. *Form Input* Animasi Metode *Selective-Reject* ARQ.
- g. *Form* Animasi Metode *Selective-Reject* ARQ.
- h. *Form* Metode ARQ
- i. *Form About*.

4.2.1. Form Utama

Fungsi *form* ini adalah sebagai tampilan utama animasi pembelajaran. Di dalam *form* ini, *user* dapat memilih metode ARQ yang akan dianimasikan.



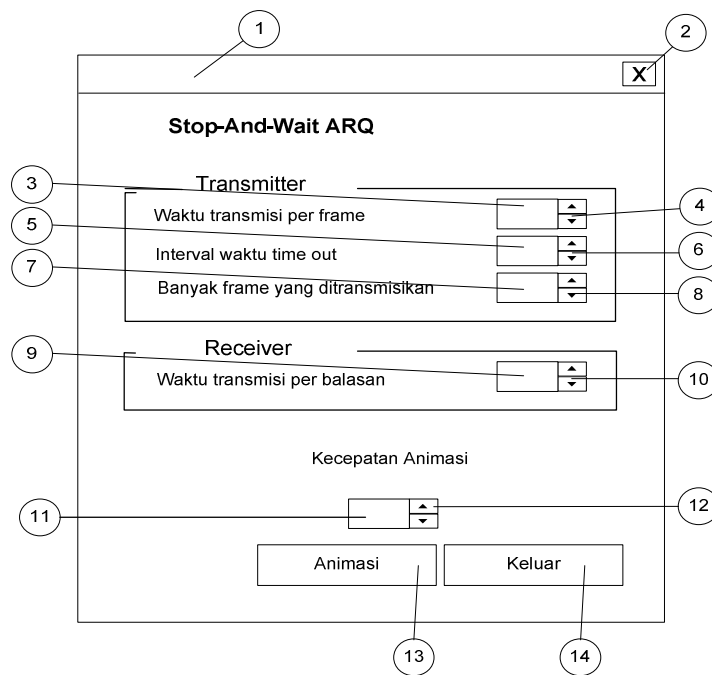
Gambar 4.15. Rancangan *form* utama

Keterangan :

- 1 : *Title bar*
- 2 : tombol 'close'.
- 3 : menu proses animasi, untuk memilih metode *stop-and-wait* ARQ, *go-back-n* ARQ, dan *selective-reject* ARQ.
- 4 : menu about, untuk memilih informasi metode ARQ dan informasi programmer.
- 5 : gambar logo.

4.2.2. Form Input Animasi Metode Stop-and-Wait ARQ

Fungsi *form* ini adalah sebagai parameter *input-an* untuk proses animasi metode *Stop-and-Wait ARQ*.



Gambar 4.16. Rancangan *form input* animasi metode *stop-and-wait ARQ*

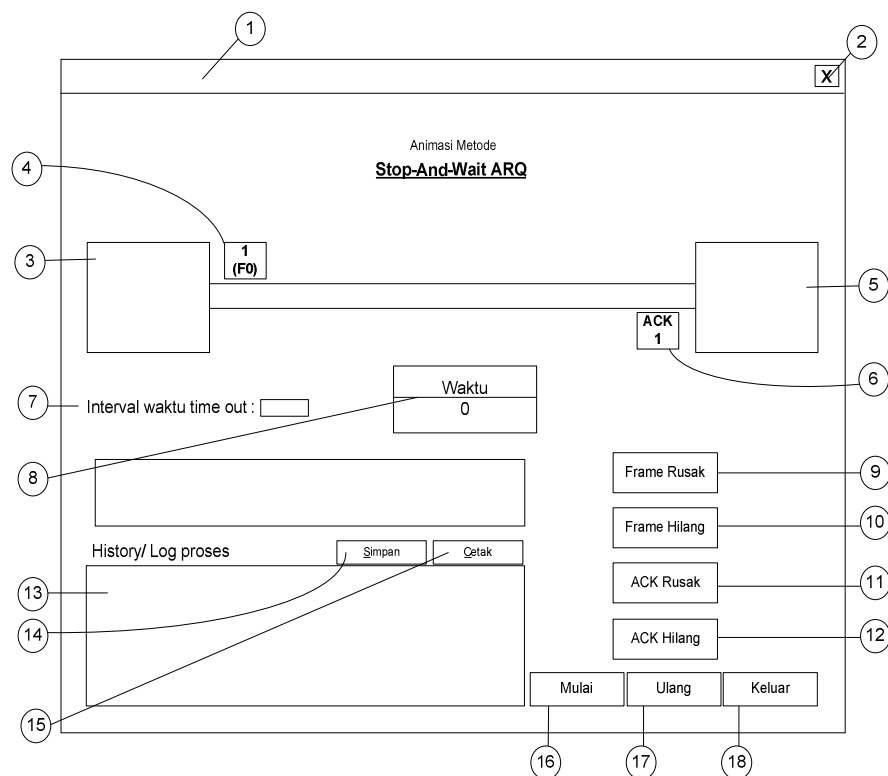
Keterangan :

- 1 : *Title bar*
- 2 : tombol 'close'.
- 3 : *textbox*, berfungsi untuk menampilkan waktu transmisi per frame.
- 4 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 3).
- 5 : *textbox*, berfungsi untuk menampilkan interval waktu 'time out'.
- 6 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 5).
- 7 : *textbox*, untuk menampilkan banyak frame yang akan ditransmisikan.
- 8 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 7).

- 9 : *textbox*, berfungsi untuk menampilkan waktu transmisi per ACK.
- 10 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 9).
- 11 : *textbox*, berfungsi untuk menampilkan nilai kecepatan .
- 12 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 11).
- 13 : tombol ‘animasi’, berfungsi untuk melanjutkan proses ke *form* .
- 14 : tombol ‘keluar’, berfungsi untuk menutup *form*.

4.2.3. Form Animasi Metode Stop-and-Wait ARQ

Fungsi *form* ini adalah untuk menampilkan animasi proses pengontrolan kesalahan dengan metode *Stop-and-Wait* ARQ.



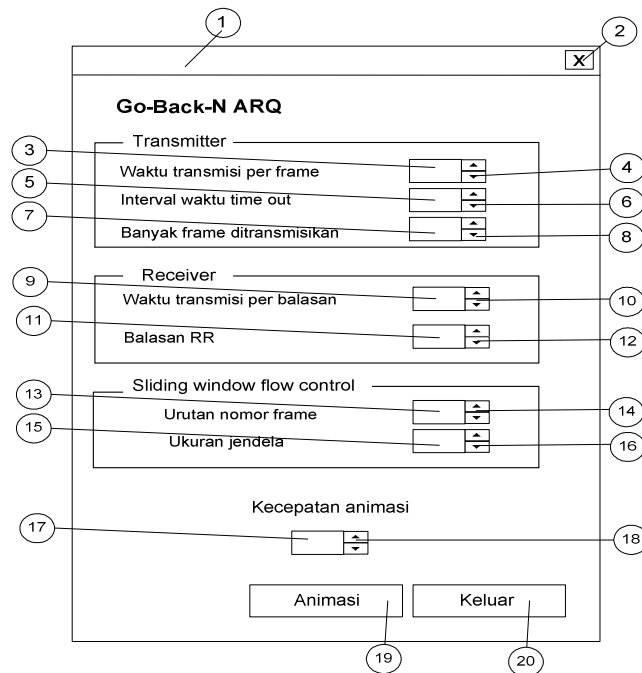
Gambar 4.17. Rancangan *form* animasi metode *stop-and-wait* ARQ

Keterangan :

- 1 : *Title bar*.
- 2 : tombol 'close'.
- 3 : *picturebox*, sebagai objek gambar komputer yang mengirimkan data.
- 4 : *picturebox*, sebagai objek gambar frame yang dikirimkan.
- 5 : *picturebox*, sebagai objek gambar komputer yang menerima data.
- 6 : *picturebox*, sebagai objek gambar ACK (balasan) yang dikirimkan.
- 7 : *label*, menunjukkan interval waktu 'time out'.
- 8 : *label*, menunjukkan waktu pada proses .
- 9 : tombol, untuk membuat gangguan frame yang rusak.
- 10 : tombol, untuk membuat gangguan frame yang hilang.
- 11 : tombol, untuk membuat gangguan ACK (balasan) yang rusak.
- 12 : tombol, untuk membuat gangguan ACK (balasan) yang hilang.
- 13 : *textbox*, untuk menunjukkan *history / log* proses.
- 14 : tombol 'simpan', untuk menyimpan *history / log* proses.
- 15 : tombol 'cetak', untuk mencetak *history / log* proses.
- 16 : tombol 'mulai/hentikan', untuk memulai atau menghentikan proses animasi.
- 17 : tombol 'ulang', untuk mengulangi (*reset*) proses animasi.
- 18 : tombol 'keluar', untuk menutup *form*.

4.2.4. Form Input Animasi Metode Go-back-N ARQ

Fungsi *form* ini adalah sebagai parameter *input-an* untuk proses animasi metode *Go-back-N ARQ*.



Gambar 4.18. Rancangan *form input* animasi metode *go-back-n* ARQ

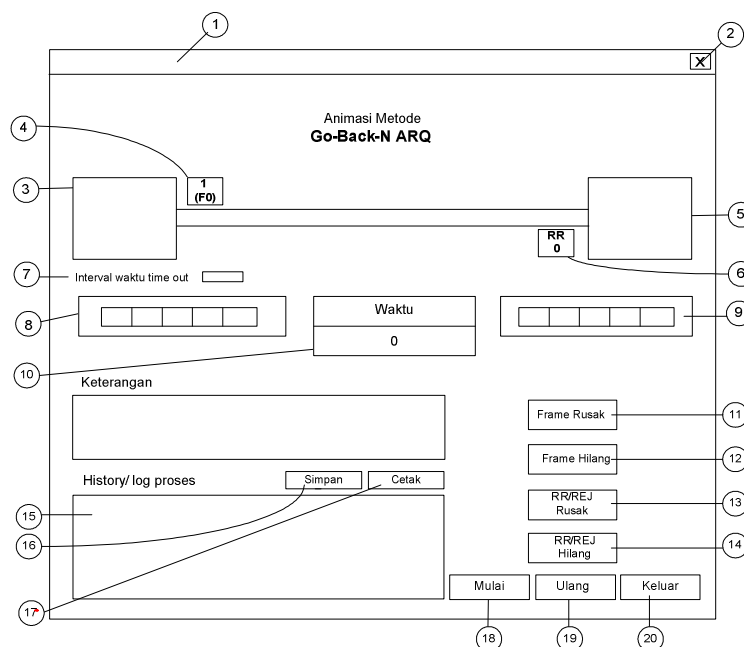
Keterangan :

- 1 : *Title bar*.
- 2 : tombol 'close'.
- 3 : *textbox*, berfungsi untuk menampilkan waktu transmisi per frame.
- 4 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 3).
- 5 : *textbox*, berfungsi untuk menampilkan interval waktu 'time out'.
- 6 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 5).
- 7 : *textbox*, untuk menampilkan banyak frame yang akan ditransmisikan.
- 8 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 7).
- 9 : *textbox*, berfungsi untuk menampilkan waktu transmisi per balasan.
- 10 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 9).
- 11 : *textbox*, untuk menampilkan banyak frame RR (*receive ready*)

- 12 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 11).
- 13 : *textbox*, berfungsi untuk menampilkan *bit* urutan nomor frame.
- 14 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 13).
- 15 : *textbox*, berfungsi untuk menampilkan besar ukuran jendela pergeseran.
- 16 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 15).
- 17 : *textbox*, berfungsi untuk menampilkan nilai kecepatan .
- 18 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 18).
- 19 : tombol ‘animasi’, berfungsi untuk melanjutkan proses ke *form* animasi.
- 20 : tombol ‘keluar’, berfungsi untuk menutup *form*.

4.2.5. Form Metode Go-back-N ARQ

Fungsi *form* ini adalah untuk menampilkan proses animasi pengontrolan kesalahan dengan metode *Go-back-N ARQ*.



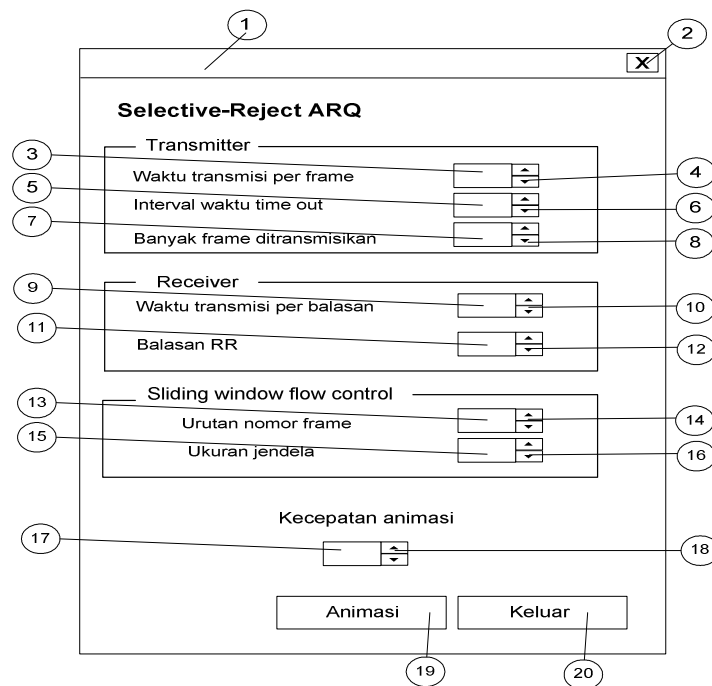
Gambar 4.19. Rancangan *form* animasi metode *go-back-n ARQ*

Keterangan :

- 1 : *Title bar*.
- 2 : tombol 'close'.
- 3 : *picturebox*, sebagai objek gambar komputer yang mengirimkan data.
- 4 : *picturebox*, sebagai objek gambar frame yang dikirimkan.
- 5 : *picturebox*, sebagai objek gambar komputer yang menerima data.
- 6 : *picturebox*, sebagai objek gambar ACK (balasan) yang dikirimkan.
- 7 : *label*, menunjukkan interval waktu 'time out'.
- 8 : objek gambar *sliding window flow control* pada *transmitter*.
- 9 : objek gambar *sliding window flow control* pada *receiver*.
- 10 : *label*, menunjukkan waktu pada proses .
- 11 : tombol, untuk membuat gangguan frame yang rusak.
- 12 : tombol, untuk membuat gangguan frame yang hilang.
- 13 : tombol, untuk membuat gangguan berupa balasan yang rusak.
- 14 : tombol, untuk membuat gangguan berupa balasan yang hilang.
- 15 : *textbox*, untuk menunjukkan *history / log* proses.
- 16 : tombol 'simpan', untuk menyimpan *history / log* proses.
- 17 : tombol 'cetak', untuk mencetak *history / log* proses.
- 18 : tombol 'mulai/hentikan', untuk memulai atau menghentikan proses animasi.
- 19 : tombol 'ulang', untuk mengulangi (*reset*) proses animasi.
- 20 : tombol 'keluar', untuk menutup *form*.

4.2.6. Form Input Animasi Metode Selective-Reject ARQ

Fungsi *form* ini adalah sebagai parameter *input*-an untuk proses animasi metode *Selective-Reject ARQ*.



Gambar 4.20. Rancangan *form input* animasi metode *selective-reject ARQ*

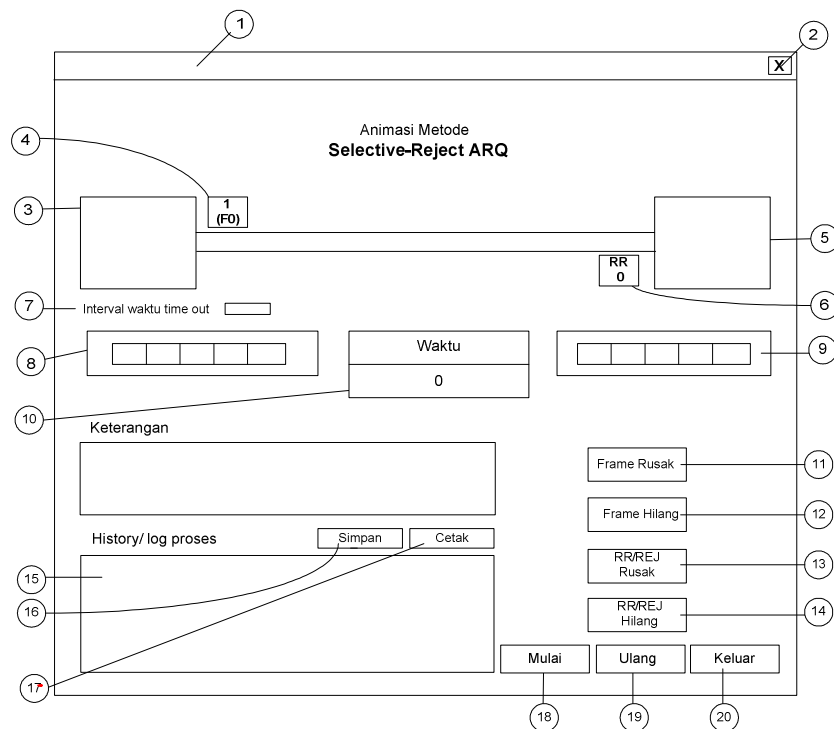
Keterangan :

- 1 : *Title bar*.
- 2 : tombol 'close'.
- 3 : *textbox*, berfungsi untuk menampilkan waktu transmisi per frame.
- 4 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 3).
- 5 : *textbox*, berfungsi untuk menampilkan interval waktu 'time out'.
- 6 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 5).
- 7 : *textbox*, untuk menampilkan banyak frame yang akan ditransmisikan.
- 8 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 7).

- 9 : *textbox*, berfungsi untuk menampilkan waktu transmisi per balasan.
- 10 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 9).
- 11 : *textbox*, berfungsi untuk menampilkan banyak frame RR (*Receive Ready*).
- 12 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 11).
- 13 : *textbox*, berfungsi untuk menampilkan *bit* urutan nomor frame.
- 14 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 13).
- 15 : *textbox*, berfungsi untuk menampilkan besar ukuran jendela pergeseran.
- 16 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 15).
- 17 : *textbox*, berfungsi untuk menampilkan perwakilan 1 *tick* dalam program dengan waktu sebenarnya (dalam milisekon).
- 18 : *updown*, berfungsi untuk mengubah nilai pada *textbox* (daerah 18).
- 19 : tombol ‘animasi’, berfungsi untuk melanjutkan proses ke *form* animasi.
- 20 : tombol ‘keluar’, berfungsi untuk menutup *form*.

4.2.7. Form Animasi Metode *Selective-Reject* ARQ

Fungsi *form* ini adalah untuk menampilkan proses animasi pengontrolan kesalahan dengan metode *Selective-Reject* ARQ. *Form* ini dapat dilihat pada gambar 4.21 berikut



Gambar 4.21. Rancangan form animasi metode selective-reject ARQ

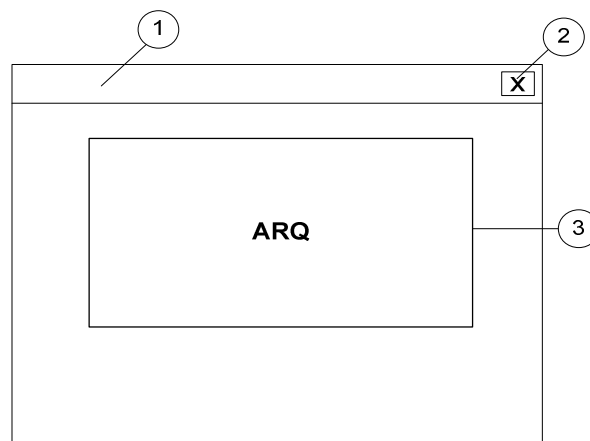
Keterangan :

- 1 : *Title bar*.
- 2 : tombol 'close'.
- 3 : *picturebox*, sebagai objek gambar komputer yang mengirimkan data.
- 4 : *picturebox*, sebagai objek gambar frame yang dikirimkan.
- 5 : *picturebox*, sebagai objek gambar komputer yang menerima data.
- 6 : *picturebox*, sebagai objek gambar ACK (balasan) yang dikirimkan.
- 7 : *label*, menunjukkan interval waktu 'time out'.
- 8 : objek gambar *sliding window flow control* pada *transmitter*.
- 9 : objek gambar *sliding window flow control* pada *receiver*.
- 10 : *label*, menunjukkan waktu pada proses .

- 11 : tombol, untuk membuat gangguan berupa frame yang rusak.
- 12 : tombol, untuk membuat gangguan berupa frame yang hilang.
- 13 : tombol, untuk membuat gangguan berupa balasan yang rusak.
- 14 : tombol, untuk membuat gangguan berupa balasan yang hilang.
- 15 : *textbox*, untuk menunjukkan *history / log* proses.
- 16 : tombol ‘simpan’, untuk menyimpan *history / log* proses.
- 17 : tombol ‘cetak’, untuk mencetak *history / log* proses.
- 18 : tombol ‘mulai/hentikan’, untuk memulai atau menghentikan proses animasi.
- 19 : tombol ‘ulang’, untuk mengulangi (*reset*) proses animasi.
- 20 : tombol ‘keluar’, untuk menutup *form*.

4.2.8. *Form* Metode ARQ

Fungsi *form* ini adalah untuk menampilkan keterangan tentang teori metode *Automatic Repeat Request* (ARQ) dalam pengontrolan kesalahan pengiriman data.



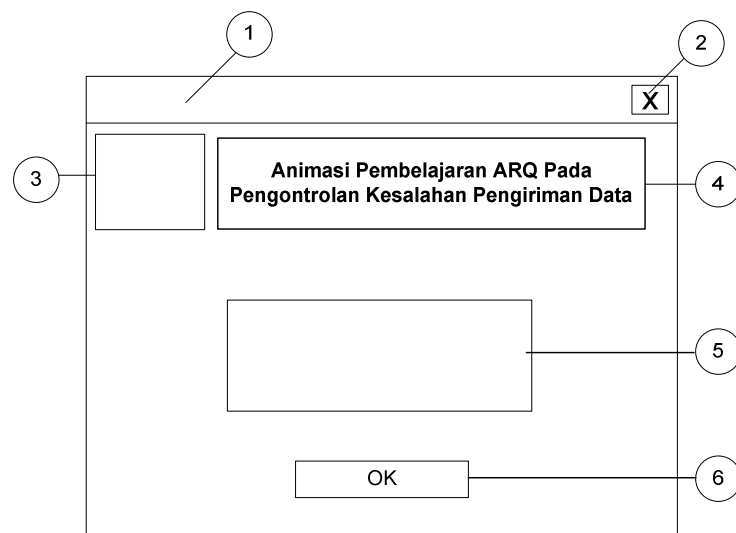
Gambar 4.22. Rancangan *form* metode ARQ

Keterangan :

- 1 : *Title bar*.
- 2 : tombol 'close'.
- 3 : keterangan tentang teori metode ARQ

4.2.9. *Form About*

Fungsi *form* ini adalah untuk menampilkan informasi mengenai pembuat animasi pembelajaran.



Gambar 4.23. Rancangan *form about*

Keterangan :

- 1 : *Title bar*.
- 2 : tombol 'close'.
- 3 : logo.
- 4 : nama animasi.
- 5 : informasi pembuat.
- 6 : tombol 'OK', untuk menutup *form*

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1. Implementasi

Implementasi merupakan tahap sistem yang siap untuk dioperasikan, termasuk kegiatan penulisan kode program yang digunakan. Implementasi animasi dari model pembelajaran virtual ini dibuat dan dibangun dengan bantuan bahasa pemrograman *Visual basic 6.0*.

5.1.1. Lingkungan Implementasi

Animasi pembelajaran ini dilakukan dalam lingkungan implementasi yang terdiri atas perangkat keras dan perangkat lunak, yaitu:

Perangkat Keras, yaitu komputer dengan spesifikasi

1. *Processor* minimum setara *pentium 3*
2. *Memory* minimum 128 MB
3. *Harddisk* minimum 20 GB

Perangkat Lunak

1. Sistem Operasi : Microsoft Windows XP Profesional
2. Bahasa Pemrograman : Microsoft Visual Basic 6.0

5.1.2. Batasan Implementasi

Batasan implementasi pada penulisan tugas akhir ini sebagai berikut:

- a. Input dari perangkat lunak animasi adalah:
 1. Waktu transmisi per frame dan waktu transmisi per balasan memerlukan nilai minimal 2 tick dan maksimal 100 tick.

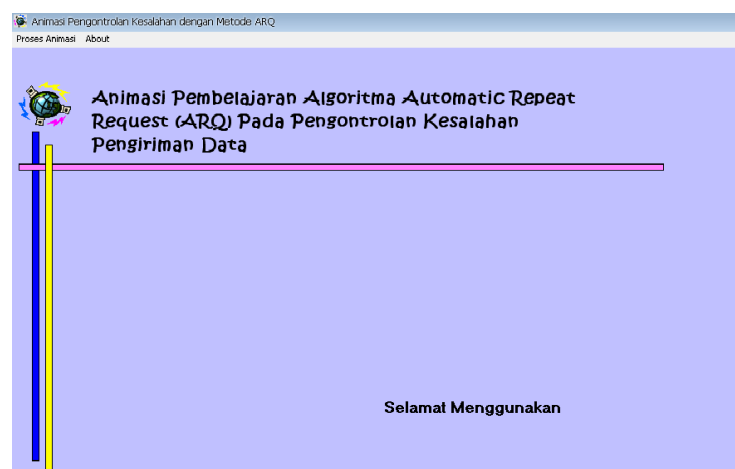
2. Interval *time out* dalam pengiriman frame minimal 13 tick.
 3. Banyak frame yang ditransmisikan maksimal 50 frame.
 4. Urutan no frame pada jendela penggeseran maksimal 5 bit.
- b. Kecepatan perambatan frame pada medium (lambat, sedang, cepat) dapat dipilih oleh *user*.

5.1.3. Implementasi Antar Muka

Implementasi antar muka ditujukan untuk mewujudkan rancangan antar muka yang telah dibuat pada bab 4 ke dalam sebuah antar muka yang dibentuk dari bahasa pemrograman Microsoft visual basic 6.0. Berikut ini implementasinya :

- a. Tampilan *form* utama

Tampilan *form* utama merupakan *form* awal yang tampil ketika perangkat lunak animasi pembelajaran dijalankan. Pada *form* ini terdiri dari menu proses animasi dan menu *about*. Pada menu proses animasi berisikan 3 pilihan proses animasi yaitu *stop-and-wait ARQ*, *go-back-n ARQ*, dan *selective reject ARQ*. Selanjutnya pada menu *about* berisikan tentang teori metode ARQ dan pembuat tugas akhir. Tampilan *form* utama dapat dilihat pada gambar 5.1 berikut.



Gambar 5.1. *Form* utama animasi pembelajaran ARQ

b. *Form input animasi metode stop-and wait ARQ*

Form input animasi ini merupakan form yang berfungsi untuk mengatur input dari tampilan animasi metode stop-and-wait ARQ yang berisikan waktu transmisi per frame, interval waktu time out, banyak frame yang ditransmisikan, waktu transmisi per balasan, dan kecepatan animasi. Tampilan form ini dapat dilihat pada gambar 5.2 berikut.

The image shows a software window titled "Metode Stop-and-Wait ARQ". Inside the window, the title "Stop-and-Wait ARQ" is displayed. The window is organized into three main sections:

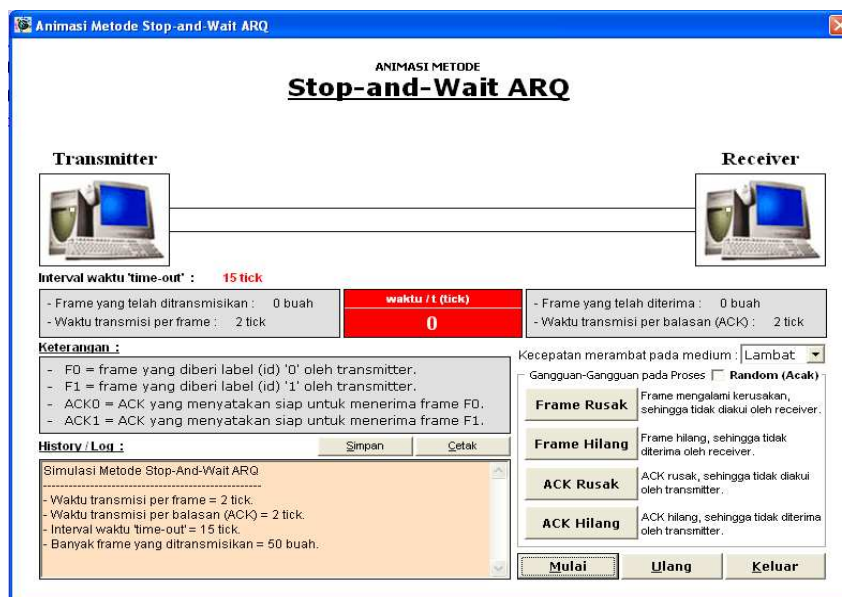
- Transmitter (SUMBER):** This section contains three input fields:
 - "Waktu transmisi per frame (tick)" with a value of 2.
 - "Interval waktu 'time-out' (tick)" with a value of 15.
 - "Banyak frame yang ditransmisikan" with a value of 5.
- Receiver (TUJUAN):** This section contains one input field:
 - "Waktu transmisi per balasan (tick)" with a value of 2.
- Kecepatan Animasi:** This section contains one input field:
 - "1 (satu) tick di dalam program = 500 milisekon waktu sebenarnya."

At the bottom of the window, there are two buttons: "Animasi" and "Keluar".

Gambar 5.2. *Form input animasi metode stop-and-wait ARQ*

c. *Form animasi metode stop-and-wait ARQ*

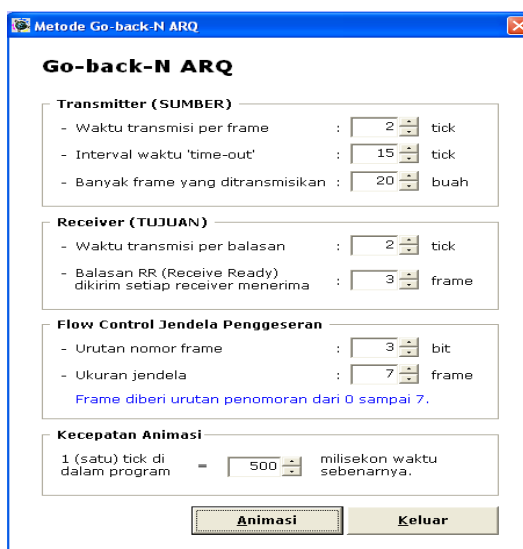
Pada *form* ini berfungsi untuk menampilkan animasi pembelajaran pada proses pengontrolan kesalahan dengan metode *stop-and-wait ARQ*. Tampilan *form* animasinya dapat dilihat pada gambar 5.3 berikut.



Gambar 5.3. Form animasi metode *stop-and-wait* ARQ

d. Form input animasi metode *go-back-n* ARQ

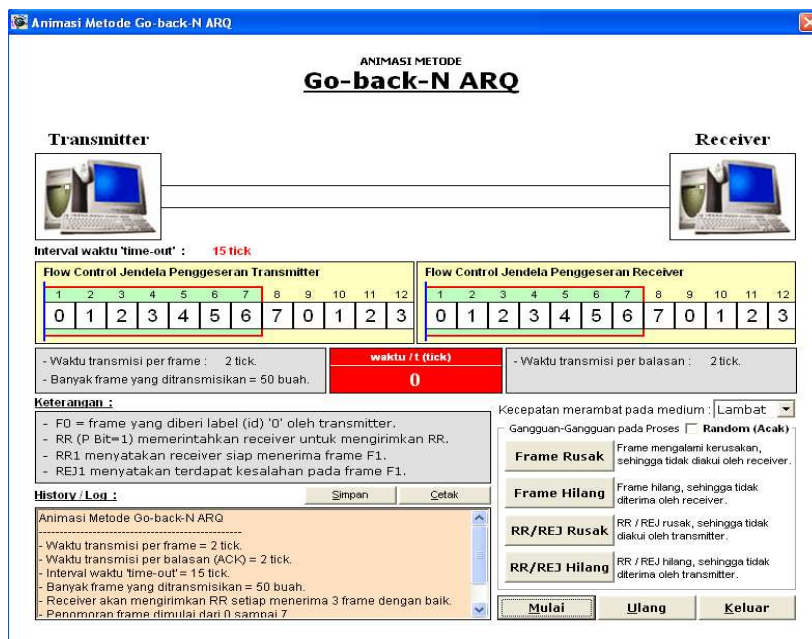
Fungsi *form input* ini untuk tampilan proses animasi metode *go-back-n*-ARQ yang berisikan waktu transmisi per frame dan per balasan, interval waktu *time out*, banyak frame yang ditransmisikan, Urutan no bit dan ukuran jendela, serta kecepatan animasi. Tampilan *form* ini dapat dilihat pada gambar 5.4 berikut.



Gambar 5.4. Form input animasi metode *go-back-n* ARQ

e. *Form animasi metode go-back-n ARQ*

Pada *form* ini berfungsi untuk menampilkan animasi pembelajaran pada proses pengontrolan kesalahan dengan metode *go-back-n ARQ*. Tampilan *form* animasinya dapat dilihat pada gambar 5.5 berikut.



Gambar 5.5. *Form animasi metode go-back-n ARQ*

f. *Form input animasi metode selective-reject ARQ*

Fungsi *form input* ini untuk menampilkan proses animasi metode *selective-reject ARQ* yang berisikan waktu transmisi per frame dan per balasan, interval waktu *time out*, banyak frame yang ditransmisikan, urutan no bit dan ukuran jendela, serta kecepatan animasi. Tampilan *form* ini dapat dilihat pada gambar 5.5 berikut.

Gambar 5.6. *Form input* animasi metode *selective-reject ARQ*

g. *Form animasi metode selective-reject ARQ*

Pada *form* ini berfungsi untuk menampilkan animasi pembelajaran pada proses pengontrolan kesalahan dengan metode *selective-reject ARQ*. Tampilan *form* animasinya dapat dilihat pada gambar 5.3 berikut

Gambar 5.7. *Form animasi metode selective-reject ARQ*

5.2. Pengujian

5.2.1. Contoh *Input Data* Animasi Metode *Stop-and-Wait* ARQ.

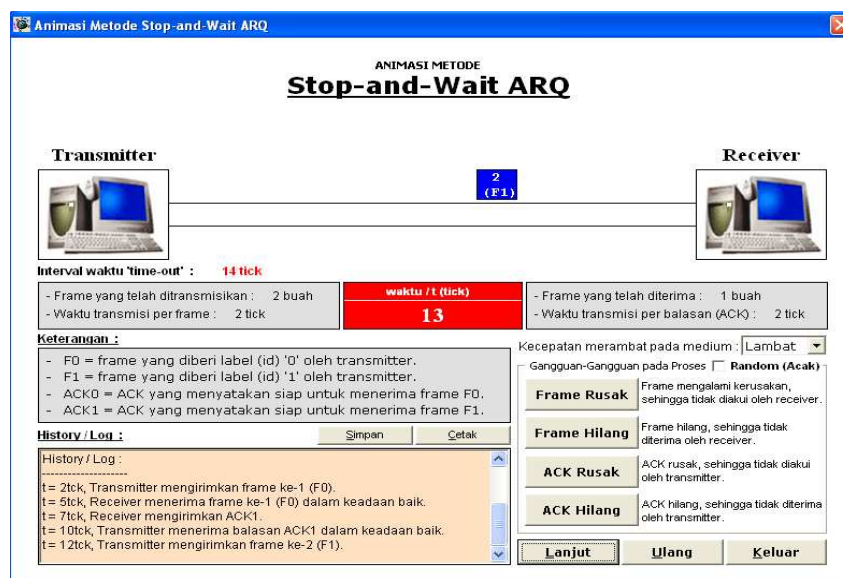
Waktu transmisi per frame = 2 tick.

Interval waktu *time-out* = 15 tick.

Banyak frame yang ditransmisikan = 15 buah.

Waktu transmisi per balasan = 2 tick.

1 tick di dalam program = 500 milisekon waktu sebenarnya



Gambar 5.8. Tampilan *form* proses animasi metode *stop-and-wait* ARQ.

History / log yang berhasil dicatat adalah sebagai berikut:

Simulasi Metode Stop-And-Wait ARQ

-
- Waktu transmisi per frame = 2 tick.
 - Waktu transmisi per balasan (ACK) = 2 tick.
 - Interval waktu 'time-out' = 15 tick.
 - Banyak frame yang ditransmisikan = 15 buah.

History / Log :

t = 2tck, Transmitter mengirimkan frame ke-1 (F0).
 t = 5tck, Receiver menerima frame ke-1 (F0) dalam keadaan baik.
 t = 7tck, Receiver mengirimkan ACK1.
 t = 10tck, Transmitter menerima balasan ACK1 dalam keadaan baik.

t = 12tck, Transmitter mengirimkan frame ke-2 (F1).
t = 15tck, Receiver menerima frame ke-2 (F1) dalam keadaan baik.
t = 17tck, Receiver mengirimkan ACK0.
t = 20tck, Transmitter menerima balasan ACK0 dalam keadaan baik.
t = 22tck, Transmitter mengirimkan frame ke-3 (F0).
t = 25tck, Frame ke-3 (F0) mengalami kerusakan, sehingga tidak diakui dan tidak diterima oleh receiver.
t = 27tck, Receiver mengirimkan NACK0.
t = 30tck, Transmitter menerima balasan NACK0 dalam keadaan baik.
t = 32tck, Transmitter mengirimkan frame ke-3 (F0).
t = 35tck, Receiver menerima frame ke-3 (F0) dalam keadaan baik.
t = 37tck, Receiver mengirimkan ACK1.
t = 40tck, Transmitter menerima balasan ACK1 dalam keadaan baik.
t = 42tck, Transmitter mengirimkan frame ke-4 (F1).
t = 45tck, Receiver menerima frame ke-4 (F1) dalam keadaan baik.
t = 47tck, Receiver mengirimkan ACK0.
t = 50tck, Transmitter menerima balasan ACK0 dalam keadaan baik.
t = 52tck, Transmitter mengirimkan frame ke-5 (F0).
t = 55tck, Receiver menerima frame ke-5 (F0) dalam keadaan baik.
t = 57tck, Receiver mengirimkan ACK1.
t = 60tck, Transmitter menerima balasan ACK1 dalam keadaan baik.
t = 62tck, Transmitter mengirimkan frame ke-6 (F1).
t = 65tck, Receiver menerima frame ke-6 (F1) dalam keadaan baik.
t = 67tck, Receiver mengirimkan ACK0.
t = 70tck, Transmitter menerima balasan ACK0 dalam keadaan baik.
t = 72tck, Transmitter mengirimkan frame ke-7 (F0).
t = 75tck, Receiver menerima frame ke-7 (F0) dalam keadaan baik.
t = 77tck, Receiver mengirimkan ACK1.
t = 80tck, Transmitter menerima balasan ACK1 dalam keadaan baik.
t = 82tck, Transmitter mengirimkan frame ke-8 (F1).
t = 85tck, Receiver menerima frame ke-8 (F1) dalam keadaan baik.
t = 87tck, Receiver mengirimkan ACK0.
t = 90tck, Transmitter menerima balasan ACK0 dalam keadaan baik.
t = 92tck, Transmitter mengirimkan frame ke-9 (F0).
t = 107tck, Waktu pada pencatat waktu di transmitter habis (time-out). Transmitter kembali mengirimkan frame ke-9 (F0).
t = 109tck, Transmitter mengirimkan frame ke-9 (F0).
t = 112tck, Receiver menerima frame ke-9 (F0) dalam keadaan baik.
t = 114tck, Receiver mengirimkan ACK1.
t = 117tck, Transmitter menerima balasan ACK1 dalam keadaan baik.
t = 119tck, Transmitter mengirimkan frame ke-10 (F1).
t = 122tck, Receiver menerima frame ke-10 (F1) dalam keadaan baik.
t = 124tck, Receiver mengirimkan ACK0.
t = 127tck, Transmitter menerima balasan ACK0 dalam keadaan baik.
t = 129tck, Transmitter mengirimkan frame ke-11 (F0).
t = 132tck, Receiver menerima frame ke-11 (F0) dalam keadaan baik.
t = 134tck, Receiver mengirimkan ACK1.
t = 137tck, Transmitter menerima balasan ACK1 dalam keadaan baik.
t = 139tck, Transmitter mengirimkan frame ke-12 (F1).
t = 142tck, Receiver menerima frame ke-12 (F1) dalam keadaan baik.
t = 144tck, Receiver mengirimkan ACK0.
t = 147tck, Transmitter menerima balasan ACK0 dalam keadaan baik.
t = 149tck, Transmitter mengirimkan frame ke-13 (F0).
t = 152tck, Receiver menerima frame ke-13 (F0) dalam keadaan baik.
t = 154tck, Receiver mengirimkan ACK1.
t = 157tck, Transmitter menerima balasan ACK1 dalam keadaan baik.

$t = 159tck$, Transmitter mengirimkan frame ke-14 (F1).
 $t = 162tck$, Receiver menerima frame ke-14 (F1) dalam keadaan baik.
 $t = 164tck$, Receiver mengirimkan ACK0.
 $t = 167tck$, Transmitter menerima balasan ACK0 dalam keadaan baik.
 $t = 169tck$, Transmitter mengirimkan frame ke-15 (F0).
 $t = 172tck$, Receiver menerima frame ke-15 (F0) dalam keadaan baik.
 $t = 174tck$, Receiver mengirimkan ACK1.
 $t = 177tck$, Transmitter menerima balasan ACK1 dalam keadaan baik.

5.2.2. Contoh Input Data Animasi Metode Go-back-N ARQ.

Waktu transmisi per frame = 2 tick.

Interval waktu *time-out* = 15 tick.

Banyak frame yang ditransmisikan = 15 buah.

Waktu transmisi per balasan = 2 tick.

Balasan RR(Receive Ready) dikirim setiap receiver menerima 3 frame.

Urutan nomor frame = 3 bit. (Penomoran frame dari 0 – 7).

Ukuran jendela = 7 frame.

1 tick di dalam program = 500 milisekon waktu sebenarnya.

ANIMASI METODE
Go-back-N ARQ

Transmitter **Receiver**

Interval waktu "time-out" : 15 tick

Flow Control Jendela Penggeseran Transmitter												Flow Control Jendela Penggeseran Receiver											
1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9	10	11	12
0	1	2	3	4	5	6	7	0	1	2	3	0	1	2	3	4	5	6	7	0	1	2	3

- Waktu transmisi per frame : 2 tick.
- Banyak frame yang ditransmisikan = 20 buah.

waktu / t (tick)
12

- Waktu transmisi per balasan : 2 tick.

Keterangan :

- F0 = frame yang diberi label (id) '0' oleh transmitter.
- RR (P Bit=1) memerintahkan receiver untuk mengirimkan RR.
- RR1 menyatakan receiver siap menerima frame F1.
- RE1 menyatakan terdapat kesalahan pada frame F1.

Kecepatan merambat pada medium : Lambat

Gangguan-Gangguan pada Proses Random (Acak)

Frame Rusak : Frame mengalami kerusakan, sehingga tidak diakui oleh receiver.

Frame Hilang : Frame hilang, sehingga tidak diterima oleh transmitter.

RR/REJ Rusak : RR / REJ rusak, sehingga tidak diakui oleh transmitter.

RR/REJ Hilang : RR / REJ hilang, sehingga tidak diterima oleh transmitter.

History / Log :

t = 7tck, Receiver menerima frame ke-2 (F1) dalam keadaan baik.
 t = 8tck, Transmitter mengirimkan frame ke-4 (F3).
 t = 9tck, Receiver menerima frame ke-3 (F2) dalam keadaan baik.
 t = 10tck, Transmitter mengirimkan frame ke-5 (F4).
 t = 11tck, Receiver mengirimkan RR3.
 t = 11tck, Receiver menerima frame ke-4 (F3) dalam keadaan baik.
 t = 12tck, Transmitter mengirimkan frame ke-6 (F5).

Gambar.5.9. Tampilan form proses animasi metode go-back-n ARQ.

History / log yang berhasil dicatat adalah sebagai berikut:

Animasi Metode Go-back-N ARQ

-
- Waktu transmisi per frame = 2 tick.
 - Waktu transmisi per balasan (ACK) = 2 tick.
 - Interval waktu 'time-out' = 15 tick.
 - Banyak frame yang ditransmisikan = 15 buah.
 - Receiver akan mengirimkan RR setiap menerima 3 frame dengan baik.
 - Penomoran frame dimulai dari 0 sampai 7.
 - Ukuran jendela = 7 frame.

History / Log :

t = 2tck, Transmitter mengirimkan frame ke-1 (F0).
t = 4tck, Transmitter mengirimkan frame ke-2 (F1).
t = 5tck, Receiver menerima frame ke-1 (F0) dalam keadaan baik.
t = 6tck, Transmitter mengirimkan frame ke-3 (F2).
t = 6tck, Receiver menerima frame ke-2 (F1) dalam keadaan baik.
t = 8tck, Frame ke-3 (F2) mengalami kerusakan, sehingga tidak diakui dan tidak diterima oleh receiver.
t = 8tck, Transmitter mengirimkan frame ke-4 (F3).
t = 10tck, Transmitter mengirimkan frame ke-5 (F4).
t = 10tck, Receiver mengirimkan REJ2.
t = 11tck, Receiver menerima frame ke-4 (F3). Frame ini tidak sesuai dengan frame yang diharapkan, sehingga dibuang oleh receiver.
t = 12tck, Transmitter mengirimkan frame ke-6 (F5).
t = 13tck, Receiver menerima frame ke-5 (F4). Frame ini tidak sesuai dengan frame yang diharapkan, sehingga dibuang oleh receiver.
t = 13tck, Transmitter menerima REJ2 dan mengirim kembali frame ke-3 (F2).
t = 15tck, Transmitter mengirimkan frame ke-3 (F2).
t = 15tck, Receiver menerima frame ke-6 (F5). Frame ini tidak sesuai dengan frame yang diharapkan, sehingga dibuang oleh receiver.
t = 17tck, Transmitter mengirimkan frame ke-4 (F3).
t = 18tck, Receiver menerima frame ke-3 (F2) dalam keadaan baik.
t = 19tck, Transmitter mengirimkan frame ke-5 (F4).
t = 20tck, Receiver menerima frame ke-4 (F3) dalam keadaan baik.
t = 21tck, Transmitter mengirimkan frame ke-6 (F5).
t = 22tck, Receiver menerima frame ke-5 (F4) dalam keadaan baik.
t = 23tck, Transmitter mengirimkan frame ke-7 (F6).
t = 24tck, Receiver mengirimkan RR5.
t = 24tck, Receiver menerima frame ke-6 (F5) dalam keadaan baik.
t = 26tck, Receiver menerima frame ke-7 (F6) dalam keadaan baik.
t = 27tck, Transmitter menerima RR5 dalam keadaan baik.
t = 29tck, Transmitter mengirimkan frame ke-8 (F7).
t = 31tck, Transmitter mengirimkan frame ke-9 (F0).
t = 32tck, Receiver menerima frame ke-8 (F7) dalam keadaan baik.
t = 33tck, Transmitter mengirimkan frame ke-10 (F1).
t = 34tck, Receiver mengirimkan RR0.
t = 35tck, Transmitter mengirimkan frame ke-11 (F2).

t = 36tck, Receiver menerima frame ke-10 (F1). Frame ini tidak sesuai dengan frame yang diharapkan, sehingga dibuang oleh receiver.
 t = 37tck, Transmitter mengirimkan frame ke-12 (F3).
 t = 37tck, Transmitter menerima RR0 dalam keadaan baik.
 t = 38tck, Receiver mengirimkan REJ0.
 t = 38tck, Receiver menerima frame ke-11 (F2). Frame ini tidak sesuai dengan frame yang diharapkan, sehingga dibuang oleh receiver.
 t = 39tck, Transmitter mengirimkan frame ke-13 (F4).
 t = 40tck, Receiver menerima frame ke-12 (F3). Frame ini tidak sesuai dengan frame yang diharapkan, sehingga dibuang oleh receiver.
 t = 41tck, Transmitter mengirimkan frame ke-14 (F5).
 t = 41tck, Transmitter menerima REJ0 dan mengirim kembali frame ke-9 (F0).
 t = 42tck, Receiver menerima frame ke-13 (F4). Frame ini tidak sesuai dengan frame yang diharapkan, sehingga dibuang oleh receiver.
 t = 43tck, Transmitter mengirimkan frame ke-9 (F0).
 t = 44tck, Receiver menerima frame ke-14 (F5). Frame ini tidak sesuai dengan frame yang diharapkan, sehingga dibuang oleh receiver.
 t = 45tck, Transmitter mengirimkan frame ke-10 (F1).
 t = 46tck, Receiver menerima frame ke-9 (F0) dalam keadaan baik.
 t = 47tck, Transmitter mengirimkan frame ke-11 (F2).
 t = 48tck, Receiver menerima frame ke-10 (F1) dalam keadaan baik.
 t = 49tck, Transmitter mengirimkan frame ke-12 (F3).
 t = 50tck, Receiver menerima frame ke-11 (F2) dalam keadaan baik.
 t = 51tck, Transmitter mengirimkan frame ke-13 (F4).
 t = 52tck, Receiver mengirimkan RR3.
 t = 52tck, Receiver menerima frame ke-12 (F3) dalam keadaan baik.
 t = 53tck, Transmitter mengirimkan frame ke-14 (F5).
 t = 54tck, Receiver menerima frame ke-13 (F4) dalam keadaan baik.
 t = 55tck, Transmitter mengirimkan frame ke-15 (F6).
 t = 55tck, Transmitter menerima RR3 dalam keadaan baik.
 t = 56tck, Receiver menerima frame ke-14 (F5) dalam keadaan baik.
 t = 57tck, Receiver menerima frame ke-15 (F6) dalam keadaan baik.
 t = 59tck, Receiver mengirimkan RR7.
 t = 62tck, Transmitter menerima RR7 dalam keadaan baik.

5.2.3. Contoh *Input Data Animasi Metode Selective-Reject ARQ.*

Waktu transmisi per frame = 2 tick.

Interval waktu *time-out* = 15 tick.

Banyak frame yang ditransmisikan = 15 buah.

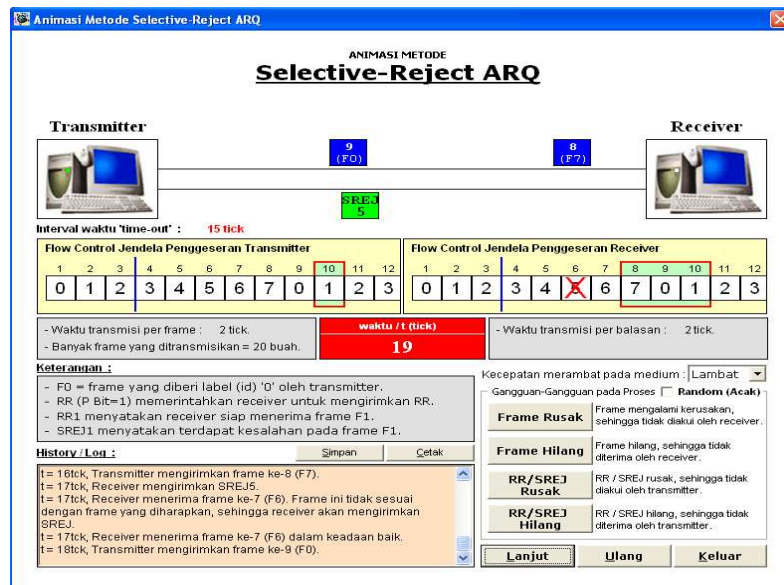
Waktu transmisi per balasan = 2 tick.

Balasan RR(*Receive Ready*) dikirim setiap *receiver* menerima 3 frame.

Urutan nomor frame = 3 bit. (Penomoran frame dari 0 – 7).

Ukuran jendela = 7 frame.

1 tick di dalam program = 500 milisekon waktu sebenarnya.



Gambar 5.10. Tampilan form proses animasi metode *Selective-reject ARQ*.

History / log yang berhasil dicatat adalah sebagai berikut:

Animasi Metode Selective-Reject ARQ

-
- Waktu transmisi per frame = 2 tick.
 - Waktu transmisi per balasan (ACK) = 2 tick.
 - Interval waktu 'time-out' = 15 tick.
 - Banyak frame yang ditransmisikan = 15 buah.
 - Receiver akan mengirimkan RR setiap menerima 3 frame dengan baik.
 - Penomoran frame dimulai dari 0 sampai 7.
 - Ukuran jendela = 7 frame.

History / Log :

-
- t = 2tk, Transmitter mengirimkan frame ke-1 (F0).
 - t = 4tk, Transmitter mengirimkan frame ke-2 (F1).
 - t = 5tk, Receiver menerima frame ke-1 (F0) dalam keadaan baik.
 - t = 6tk, Transmitter mengirimkan frame ke-3 (F2).
 - t = 7tk, Receiver menerima frame ke-2 (F1) dalam keadaan baik.
 - t = 8tk, Transmitter mengirimkan frame ke-4 (F3).

t = 8tck, Frame ke-3 (F2) mengalami kerusakan, sehingga tidak diakui dan tidak diterima oleh receiver.

t = 10tck, Transmitter mengirimkan frame ke-5 (F4).

t = 10tck, Receiver mengirimkan SREJ2.

t = 11tck, Receiver menerima frame ke-4 (F3). Frame ini tidak sesuai dengan frame yang diharapkan, sehingga receiver akan mengirimkan SREJ.

t = 11tck, Receiver menerima frame ke-4 (F3) dalam keadaan baik.

t = 12tck, Transmitter mengirimkan frame ke-6 (F5).

t = 13tck, Receiver menerima frame ke-5 (F4) dalam keadaan baik.

t = 13tck, Transmitter menerima REJ2 dan mengirim kembali frame ke-3 (F2).

t = 15tck, Transmitter mengirimkan frame ke-3 (F2) sebagai reaksi atas penerimaan balasan SREJ2.

t = 15tck, Receiver menerima frame ke-6 (F5) dalam keadaan baik.

t = 17tck, Transmitter mengirimkan frame ke-7 (F6).

t = 20tck, Receiver mengirimkan RR6.

t = 20tck, Receiver menerima frame ke-7 (F6) dalam keadaan baik.

t = 23tck, Transmitter menerima RR6 dalam keadaan baik.

t = 25tck, Transmitter mengirimkan frame ke-8 (F7).

t = 27tck, Transmitter mengirimkan frame ke-9 (F0).

t = 28tck, Receiver menerima frame ke-8 (F7) dalam keadaan baik.

t = 29tck, Transmitter mengirimkan frame ke-10 (F1).

t = 31tck, Transmitter mengirimkan frame ke-11 (F2).

t = 32tck, Receiver menerima frame ke-10 (F1). Frame ini tidak sesuai dengan frame yang diharapkan, sehingga receiver akan mengirimkan SREJ.

t = 32tck, Receiver menerima frame ke-10 (F1) dalam keadaan baik.

t = 33tck, Transmitter mengirimkan frame ke-12 (F3).

t = 34tck, Receiver mengirimkan SREJ0.

t = 34tck, Receiver menerima frame ke-11 (F2) dalam keadaan baik.

t = 35tck, Transmitter mengirimkan frame ke-13 (F4).

t = 36tck, Receiver menerima frame ke-12 (F3) dalam keadaan baik.

t = 37tck, Transmitter menerima REJ0 dan mengirim kembali frame ke-9 (F0).

t = 38tck, Receiver menerima frame ke-13 (F4) dalam keadaan baik.

t = 39tck, Transmitter mengirimkan frame ke-9 (F0) sebagai reaksi atas penerimaan balasan SREJ0.

t = 44tck, Receiver mengirimkan RR5.

t = 46tck, Transmitter menerima RR5 dalam keadaan baik.

t = 48tck, Transmitter mengirimkan frame ke-14 (F5).

t = 50tck, Transmitter mengirimkan frame ke-15 (F6).

t = 51tck, Receiver menerima frame ke-14 (F5) dalam keadaan baik.

t = 53tck, Receiver menerima frame ke-15 (F6) dalam keadaan baik.

t = 55tck, Receiver mengirimkan RR7.

t = 58tck, Transmitter menerima RR7 dalam keadaan baik.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Setelah menyelesaikan animasi pembelajaran algoritma ARQ ini, penulis menarik kesimpulan sebagai berikut:

1. Animasi pembelajaran yang dibuat, mampu menampilkan proses algoritma *Automatic Repeat Request (ARQ)* yang terdiri dari *stop-and-wait ARQ*, *go-back-n ARQ*, dan *selective-reject ARQ* pada pengontrolan kesalahan pengiriman data dalam bentuk frame.
2. Berdasarkan hasil pengujian metode *Selective-Reject ARQ* merupakan metode yang paling efisien dari sisi waktu proses, karena metode tersebut mengirimkan beberapa frame sekaligus tanpa harus menunggu balasan dari *receiver* terlebih dahulu (keunggulan dibandingkan dengan metode *Stop-and-Wait ARQ*) dan frame yang dikirim ulang hanyalah frame yang rusak atau salah (keunggulan dibandingkan dengan metode *Go-back-N ARQ*).

6.2. Saran

Penulis ingin memberikan beberapa saran yang mungkin dapat membantu dalam pengembangan pembuatan animasi pembelajaran ini yaitu :

1. Animasi pembelajaran dapat dikembangkan dengan menambahkan fasilitas suara atau gambar yang lebih menarik, sehingga diharapkan proses yang dianimasikan akan lebih mudah dimengerti.

2. Untuk mendapatkan animasi yang lebih baik, animasi pembelajaran dapat dibangun dengan menggunakan aplikasi *Macromedia Flash*.

DAFTAR PUSTAKA

- Dina, Utami , "*Animasi dalam pembelajaran*" [online] Available [http:// www..scribd.com/doc/7988195/2221242120112032007111113/efektifitsanimasi-dalam-pembelajaran](http://www.scribd.com/doc/7988195/2221242120112032007111113/efektifitsanimasi-dalam-pembelajaran), diakses 24 september 2009.
- Dony, Ariyus dan Rum, Andri KR., "*Komunikasi Data*", Andi yogyakarta, Agustus 2008.
- Endro, Wibowo, "*Belajar Lebih Menyenangkan Dengan Animasi*", [online] Available <http://www.edubenchmark.com/belajar-lebih-menyenangkan-dengan-animasi.html>, diakses 10 September 2009.
- Gigih, Adi dan Yuni, Almadin, "*SELECTIVE REPEAT*", Jurusan Teknik Elektro Fakultas Teknologi Industri Institut Teknologi Sepuluh Nopember Surabaya, 2008.
- Kristanto, Andri, "*Control Error*", *Jaringan Komputer*, Graha Ilmu, Yogyakarta, 2003.
- Kurniawan, Tjandra "*Tip Trik Unik Visual Basic*", halaman 1-8, Elex Media Komputindo, Jakarta, 2003.
- Nugroho, Purnomo, "*Animasi religiositas*" [online] Available <http://purwonomedia.wordpress.com/2008/10/24/animasi-religiositas/>, diakses 24 September 2009.
- Sarosa, Moechammad, dan Anggoro, Sigit, "*Jaringan Komputer Data Link, Network & Issue*", Teknik Sistem Komputer Elektroteknik Institut Teknologi Bandung, 2000.
- Stallings, William, "*Data Link Control*", *Komunikasi Data dan Komputer*, Edisi 1, halaman 197-212, Jakarta, Salembal Teknika, 2001.
- Tanenbaum, Andrew S., "*Model Referensi OSI*", *Jaringan Komputer (Edisi Bahasa Indonesia)*, halaman 27-32, Prehallindo, Jakarta, 2000.