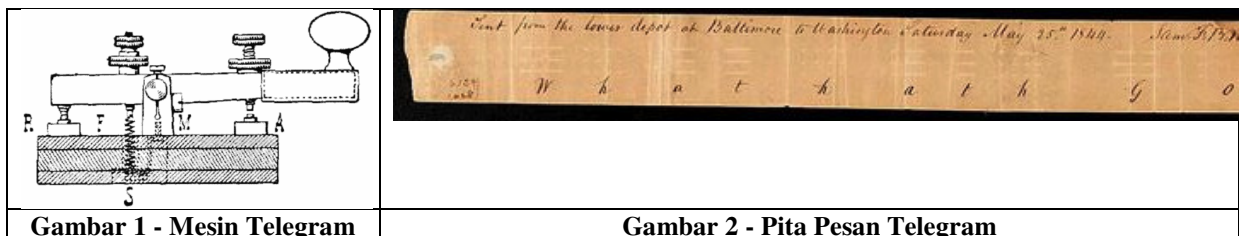


## SISTEM TELEGRAM

### Pengantar

Sebuah **sistem telegram** (*telegram system*) adalah sebuah sistem pengiriman pesan jarak jauh dimana pesan dikirimkan dengan signal listrik melalui sebuah kabel. Dalam mengirimkan pesan, sistem ini membutuhkan sebuah skema pengkodean yang dimengerti oleh pengirim dan penerima. Desain skema encoding pada sistem telegraph sangat bergantung pada kemampuan dan keterbatasan dari media pensignalan (i.e., kabel + signal listrik).

(Morse 1869) menyebutkan sistem telegram pertama ditemukan Schilling pada tahun 1832 saat dan pada tahun 1837 Morse telah berhasil melakukan percobaan mengirimkan dan merekam pesan telegram dengan sebuah mesin elektronik (**Gambar 1**). Pada mesin tersebut Morse menggunakan sebuah skema pengkodean, yang sekarang lazim disebut kode Morse, yang tercetak/terekam pada sebuah pita kertas (**Gambar 2**) yang selanjutnya akan diterjemahkan kembali dalam serangkainya alfabet.

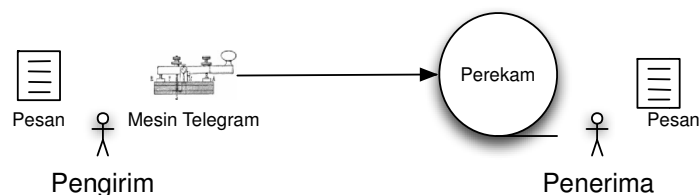


Akan tetapi kode morse hanya melakukan pengkodean terhadap simbol alfabet (i.e., a-z), sedangkan karakter lain (e.g., karakter khusus, tanda baca) haruslah direpresentasikan dalam alfabet sebelum dikirim melalui telegram. Berikut adalah sejumlah contoh representasi alfabet dari karakter lain-lain:

Karakter (non Alfabet)	Representasi Alfabet
1	satu
2	dua
!	tandaseru
@	at
...	...

Tabel 1 - Kamus Telegram

Sehingga secara umum (pada Gambar 3), sebuah pesan akan dibaca oleh operator pengirim dan dikodekan menjadi kode morse. Selanjutnya kode-kode tersebut dikirimkan melalui jaringan telegram ke penerima. Operator penerima akan menterjemahkan kode-kode morse tersebut sehingga menjadi sebuah pesan yang dapat langsung dibaca oleh penerima akhir dari pesan telegram.



Gambar 3 - Sistem Telegram

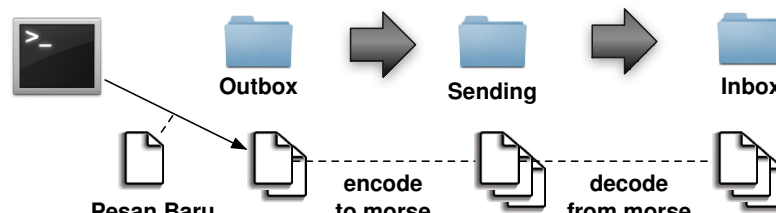
Pada umumnya, skema biaya dari sebuah telegram berdasar pada jumlah kata yang dikirimkan, dan pada umumnya kata-kata akan digolongkan menjadi sejumlah kategori harga berdasar pada panjang kata dan fungsi kata tersebut dalam pesan (e.g., tanda baca, karakter khusus, dll.), sebagai berikut:

Jenis Kata	Harga
Pendek ( $\leq 11$ karakter)	500
Panjang ( $> 11$ karakter)	800
Tanda Baca (! , . ? / \ ` " : ; )	200
Lain-Lain (~ @ # \$ % ^ & * ( ) > < - + =)	700
Enter/Baris Baru	0

**Tabel 2 - Harga Telegram per Kata**

## Simulasi Sistem Telegram

Pada umumnya setiap pesan telegram, akan dikirimkan melalui sebuah jaringan kabel/telegram. Akan tetapi pada tugas ini anda diminta membangun simulasi sistem telegram dalam Bahasa C dengan menggantikan pengiriman/penerimaan pesan-pesan tersebut menjadi pemindahan sebuah file dari suatu folder ke folder lain berdasar status pengirimannya seperti pada Gambar 4.



**Gambar 4 - Skema Simulasi Sistem Telegram**

Pada simulasi ini, sistem pada Gambar 3 akan disimulasikan menjadi sebuah sistem piranti lunak/program yang memiliki tiga modus operasi (penulisan pesan, pengiriman pesan, dan penerimaan pesan). Secara umum pengirim akan menuliskan pesan telegram pada modus penulisan pesan, dan setiap pesannya akan tersimpan pada sebuah file di folder "Outbox". Pesan tersebut baru akan dikirim jika ada program yang berjalan pada modus pengiriman pesan, aktifitas ini disimulasikan dengan cara melakukan translasi pesan yang dituliskan pada modus pengiriman pesan kedalam kode morse dan diletakkan pada folder "Sending". Akhirnya pada modus penerimaan pesan, pesan yang ada pada folder "Sending" dibaca kembali menjadi pesan tekstual seperti yang terkirim pada awalnya.

Selain dari ketiga modus operasi, simulasi ini memiliki sebuah modus tambahan, modus pengaturan. Pada modus ini pengguna dapat mendefinisikan sejumlah parameter, seperti: kamus pemetaan, harga, dan data pelanggan. Pengguna dapat berganti modus penggunaan sesuai dengan kebutuhan/*use case* saat itu.

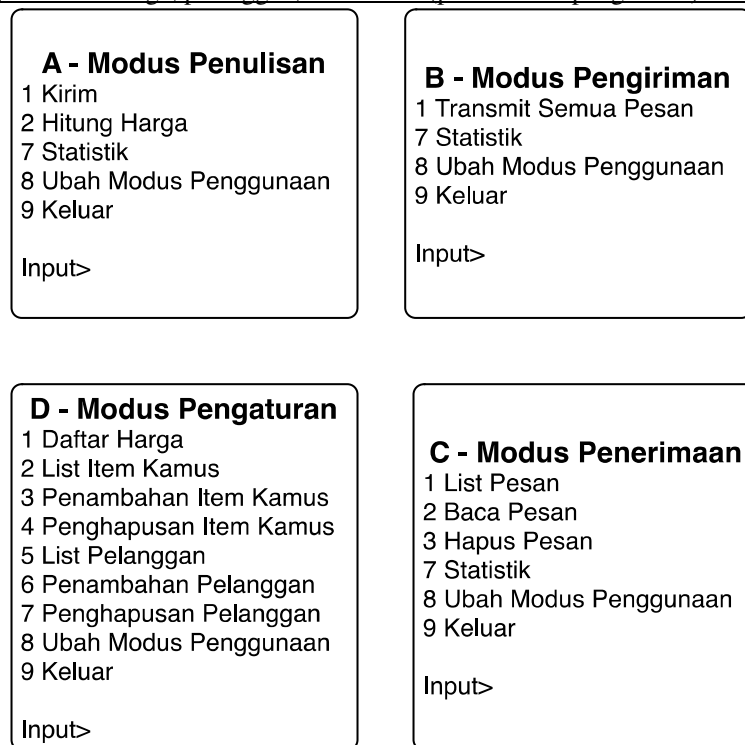
## Interaksi Program dengan Pengguna

Ketika program dipanggil pertama kali, pengguna masuk ke *command prompt* yang ditandai dengan prompt ">" dan siap melakukan operasi simulasi dari sistem telegram pada modus pengiriman pesan. Perintah-perintah di lingkungan ini diketikkan setelah prompt "INPUT>". Jika dimasukkan pilihan "8" maka program akan memberikan sejumlah pilihan menu yang dapat mengubah program ke modus lain, seperti pada menu berikut:

```
A - Modus Penulisan
B - Modus Pengiriman
C - Modus Penerimaan
D - Modus Pengaturan

Masukkan modus yang dikehendaki:
```

Struktur menu lengkapnya dapat dilihat pada Gambar 5. Modus Pengiriman, Penerimaan, dan Pengaturan hanya dapat dijalankan jika daftar harga, pelanggan, dan kamus (pada modus pengaturan) telah terdefinisi seluruhnya.



Gambar 5 - Struktur Menu

### Modus Penulisan

Modus ini berfungsi menerima masukan pesan dari operator, dan mengkonversikan ke dalam representasi yang dapat dikirimkan melalui telegram. Berikut adalah pilihan menu yang tersedia pada Modus Penulisan.

#### 1. Kirim

Menu pilihan yang digunakan untuk mengirimkan pesan dari seorang pelanggan ke pelanggan lainnya. Pada menu ini operator telegram/pengguna diminta memasukkan: nama pengirim, nama penerima, dan pesan<sup>1</sup>.

```
Prioritas      : -
Nama Pengirim  : Budi
Nama Penerima  : Iwan
Pesan          : Urgent, harap segera pulang. Ayah sakit keras.
```

Proses pengiriman ini akan sukses jika masukan pengguna telah benar.

- Nama pengirim dan penerima telah terdaftar pada sistem sebagai pelanggan
- Seluruh tanda baca dan karakter khusus telah terdefinisi di kamus

Jika hal-hal tersebut tidak terpenuhi proses pengiriman gagal dan sistem akan meminta operator untuk melakukan koreksi hingga kedua kriteria tersebut terpenuhi. Jika hingga 3 kali koreksi masukan tersebut masih belum valid maka sistem akan membatalkan pengiriman pesan ini, dan jika operator masih ingin mengirimkan pesan tersebut maka ia harus memulai dari awal.

Secara internal, menu ini akan menyimpan pesan tersebut pada folder "Outbox" dengan nama file: YYMMDDHHMMSS-<pengirim>.txt dimana <pengirim> adalah nama pengirim dan YY adalah tahun pengiriman, MM adalah bulan pengiriman, DD adalah tanggal pengiriman, HH adalah jam pengiriman, MM adalah menit pengiriman, dan SS adalah detik pengiriman. Pesan tersebut harus kedalam bentuk yang dapat dikirimkan via telegram/kode morse. Contoh hasil konversi pesan diatas dapat dilihat sebagai berikut:

<sup>1</sup> Pesan ditulis dengan case insensitive.

## Tugas Besar - IF2030 - Algoritma dan Struktur Data

### Simulasi Sistem Telegram

```
Nama Pengirim TitikDua Budi ENTER Nama Penerima TitikDua Iwan ENTER
Pesan TitikDua Urgent KOMA harap segera pulang TITIK Ayah sakit keras
TITIK
```

Setelah pesan sukses disimpan maka program mencetak ke layar jumlah kata dari masing-masing jenis kata (pada Tabel 2) dan biaya atas pesan yang baru dikirimkan oleh pengirim.

Catatan: Isian Prioritas hanya relevan jika anda berniat mengerjakan salah satu item bonus perihal *priority-queue*.

#### 2. Hitung Harga

Mencetak informasi perintah pesan terakhir yang dikirim. Informasi itu meliputi: pengirim, penerima, perhitungan harga dan jumlah kata dari masing-masing jenis kata dari pesan terakhir yang dikirim.

```
Info harga dari pesan terakhir
Nama Pengirim: Budi
Nama Penerima: Iwan
Harga 8200
Jumlah Total Kata 22 kata
    Kata Pendek 14
    Kata Panjang 0
    Tanda Baca 6
    Lain-lain 0
Enter 2
```

#### 7. Statistik

Mencetak statistik pesan yang ditulis oleh seluruh pelanggan yang pernah dikirim melalui sistem ini. Statistik pada modul ini meliputi: Nama pengirim, Jumlah Pesan yang dikirim, Rata-rata jumlah total kata dalam sebuah pesan, dan total harga dari pesan-pesan yang pernah dikirim.

```
Budi   20 Pesan   30 kata/pesan   160000
Iwan   2 Pesan   60 kata/pesan   40000
...
```

Catatan: Tidak terdapat batasan maksimum jumlah pelanggan pada statistik ini. Statistik ini harus disimpan secara persisten pada sebuah file.

#### 8. Ubah Modus Penggunaan

Menu untuk berpindah modus penggunaan dari sistem simulasi telegram.

```
A - Modus Penulisan
B - Modus Pengiriman
C - Modus Penerimaan
D - Modus Pengaturan

Masukkan modus yang dikehendaki:
```

#### 9. Keluar

Menu untuk keluar dari program sistem simulasi telegram.

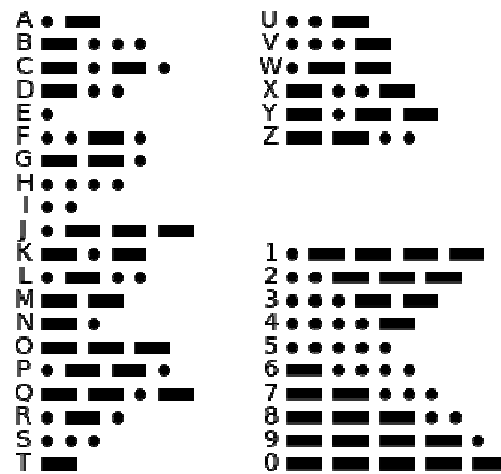
#### Modus Pengiriman

Modus penggunaan ini berfungsi untuk mengirimkan pesan yang telah terkonversikan ke pengguna. Berikut adalah pilihan menu yang tersedia pada modul ini.

##### 1. Transmisi Semua Pesan

Menu ini melakukan pengiriman seluruh pesan yang telah ditulis ke tujuan. Pada simulasi ini, pengiriman sebuah pesan melalui jaringan telegram disimulasikan sebagai tindakan mentranslasikan sebuah pesan kedalam

kode morse dan meletakkan pada folder *Sending* dengan nama file yang sama. Setelah proses tersebut sukses dilakukan, maka file pada folder *Outbox* harus dihapus.



Gambar 6 - Kode Morse

Saat menu ini dipilih, program akan memuat/me-load seluruh file-file pesan pada folder *Outbox* sebagai antrian pekerjaan yang harus dikirimkan. Antrian pekerjaan pada simulasi ini terurut berdasar waktu penulisan pesan (dapat dilihat dari nama file). Pada simulasi ini, media pengiriman hanya mampu mengirimkan **satu pesan pada tiap 30 detik**.

Catatan: Ketika menggunakan *priority-queue* (bonus), urutan antrian memperhatikan isian prioritas.<sup>2</sup> Jika terdapat sejumlah pesan yang memiliki prioritas yang sama, maka urutan akan didasarkan pada waktu penulisan pesan.

#### 7. Statistik

Mencetak statistik jumlah pengiriman pesan pada tiap jamnya.

```
2012-10-28 12      20 Pesan
2012-10-28 13       2 Pesan
2012-10-29 14       4 Pesan
...
```

Catatan: rekaman ini akan disimpan secara persisten pada sebuah file.

#### 8. Ubah Modus Penggunaan

Sama dengan menu pada modus pengiriman

#### 9. Keluar

Sama dengan menu pada modus pengiriman

#### Modus Penerimaan

##### 1. List Pesan

Menu ini menampilkan list pesan-pesan yang telah terkirim. Secara internal program simulasi melakukan konversi file dengan representasi kode morse pada folder "*Sending*" kedalam representasi tekstual (String alfabet) dan diletakkan pada folder "*Inbox*", dan jika proses ini sukses maka file pada folder "*Sending*" harus dihapus. Kemudian program akan mencetak ke layar nama file dari pesan-pesan tersebut.

```
1. 121028121300-Budi.txt
2. 121028121400-Iwan.txt
3. 121029121500-Budi.txt
...
```

<sup>2</sup> Simulasi ini menggunakan 4 kelas prioritas: Biasa, Segera, Penting, dan Kritis.

## 2. Baca Pesan

Menu ini akan membutuhkan masukan id pesan dari pengguna. Jika id terdefinisi maka pesan akan dicetak ke layar, dengan merepresentasi pesan sebagaimana pesan tersebut dikirimkan. Dengan kata lain akan mengubah kembali representasi alfabet dari tanda baca, karakter khusus, dan enter menjadi simbol-simbol semual, sehingga lebih mudah dibaca oleh pengguna.

Contoh:

```
Input> 2
Masukkan Id Pesan> 1
```

sehingga pesan<sup>3</sup>:

```
Nama Pengirim TitikDua Budi ENTER Nama Penerima TitikDua Iwan ENTER
Pesan TitikDua Urgent KOMA harap segera pulang TITIK Ayah sakit keras
TITIK
```

akan ditampilkan menjadi:

```
Nama Pengirim : Budi
Nama Penerima : Iwan
Pesan : Urgent , harap segera pulang . Ayah sakit keras .
```

## 3. Hapus Pesan

Menu ini akan membutuhkan masukan id pesan dari pengguna. Jika id terdefinisi maka pesan akan dihapus dari folder "Inbox" dan tidak akan muncul lagi pada list pesan.

Contoh:

```
Input> 3
Masukkan Id Pesan> 1
```

Pesan 121028121300-Budi.txt telah terhapus dari sistem.

## 7. Statistik

Mencetak statistik pembacaan pesan yang telah diterima. Statistik pada modus ini meliputi: Nama Penerima, Jumlah Pesan yang diterima dan Jumlah pesan yang dibaca.

```
Iwan 10/20 Pesan
Budi 2/2 Pesan
...
```

Catatan: Rekaman ini disimpan secara persisten pada sebuah file..

## 8. Ubah Modus Penggunaan

Sama dengan menu pada modus pengiriman.

## 9. Keluar

Sama dengan menu pada modus pengiriman.

## Modus Pengaturan

Seluruh data pada modus ini harus tersimpan secara persisten di file.

## 1. Daftar Harga

Menampilkan list harga dari jenis-jenis kata.

Catatan: Konfigurasi pada Tabel 2 harus *configurable* dengan mengubah harga dari sebuah jenis kata pada file konfigurasi.

---

<sup>3</sup> Perbedaan case (huruf besar/kecil) pada pesan di dokumen ini hanya digunakan untuk memudahkan pembacaan. Pesan pada file di outbox atau inbox disimpan secara case insensitive (huruf besar/kecil semua).

## 2. List Item Kamus

Mencetak ke layar daftar item yang ada di kamus, sebagai contoh sebagai berikut:

'	-	aphostroph
~	-	tilde
!	-	tandaseru
@	-	at
#	-	hash
\$	-	dollar
%	-	percentage
^	-	caret
&	-	and
*	-	asterisk
(	-	kurungbuka
)	-	kurungtutup
"	-	kutipganda

List item kamus dapat diubah berdasar

## 3. Penambahan Item Kamus

Menambah item baru ke dalam kamus, jika item tersebut belum terdefinisi ke dalam kamus.

## 4. Penghapusan Item Kamus

Menghapus item yang telah terdefinisi pada kamus.

## 5. List Pelanggan

Mencetak ke layar list pelanggan yang telah terdaftar di sistem.

## 6. Penambahan Pelanggan

Mendaftarkan pelanggan baru ke dalam sistem, jika pelanggan belum terdaftar dalam sistem.

## 7. Penghapusan Pelanggan

Menghapus entri pelanggan yang terdefinisi dalam sistem.

## 8. Ubah Modus Penggunaan

Sama dengan menu pada modus pengiriman.

## 9. Keluar

Sama dengan menu pada modus pengiriman.

## Deskripsi Pengerjaan Tugas

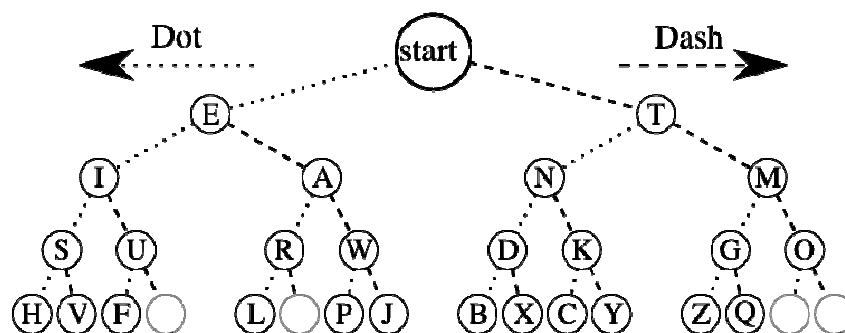
- Beberapa batasan yang berlaku untuk tugas ini (sebelum pengerjaan bonus):
  - Semua proses akuisi input dari pengguna harus dilakukan dengan mesin kata.
  - Minimum tanda baca dan simbol yang harus di dukung: ' ~ ! @ # \$ % ^ & \* ( ) \ / ? > < . , ; ; " - + =
- Tugas ini menggunakan banyak ADT dan mesin yang pernah diajarkan sampai saat tugas ini diberikan dengan modifikasi yang diperlukan.
- Tugas harus diserahkan per kelompok (maksimum 5 orang dalam satu kelas), dengan cara pengelompokan yang akan ditentukan. Setiap mahasiswa harus memahami spesifikasi, dan akan diberikan 1 sesi khusus untuk klarifikasi spesifikasi.
- Waktu dan tatacara penyerahan serta pemeriksaan tugas akan diumumkan di situs <http://kuliah.itb.ac.id> pada *link* kuliah IF2030/Algoritma dan Struktur Data.
- Pembagian tugas (penanggung jawab masing-masing modul program) harus dinyatakan dengan jelas dalam penyerahan tugas.
- Deadline: Senin, 3 Desember 2012 17.00 WIB

## Deliverables

1. *Source code, well commented*, dengan pembagian *file* yang sesuai standar ([namaADT].h, [namaADT].c, [driver].c). Primitif dalam ADT hanya primitif yang dipakai dalam program. Satu modul ADT disimpan dalam satu folder yang terpisah.
2. Laporan yang berisi antara lain:
  - Detil spesifikasi.
  - Sketsa struktur data internal.
  - Data test.
  - *Test script*, yaitu urutan tes yang “masuk akal” dan mencakup minimal semua fasilitas yang harus dites serta sesuai spesifikasi di atas.
  - Notulen setiap pertemuan untuk klarifikasi spesifikasi yang dilakukan per kelompok.Akan diberikan *template* laporan.

## Bonus

- Pengkodean dan peng-dekodean kode morse direpresentasikan secara *binary-search-tree*.



- **Gambar 7 - Pengkodean Kode Morse dalam bentuk Pohon**

- Antrian dari pengiriman pesan adalah merupakan dapat diubah menjadi *priority-queue* hanya dengan mengganti library antrian/queue pada saat proses objek linking
- Simulasi ini dapat berjalan dengan menjalankan 3 (tiga) *instance* dari program ini yang bekerja secara paralel dalam sebuah host yang sama, asalkan ketiga program tersebut berjalan dengan modus yang berbeda-beda: penulisan, pengiriman, dan penerimaan.
- Validasi untuk input perintah (*command*) maupun file.
- Sorting dari data statistik berdasarkan field tertentu.
- Penanganan khusus terhadap kata pesan yang sama dengan kata dalam kamus, misalnya: TITIK (bisa merupakan nama orang, bisa juga merupakan tanda baca yang ada di kamus).