

Spesifikasi Tugas Besar IF1210 Dasar Pemrograman

Deskripsi Persoalan

Seribu tahun yang lalu, terdapat sebuah perpustakaan legendaris di kota Ba Sing Tse. Perpustakaan tersebut rumornya dijaga oleh seekor burung hantu raksasa yang bernama Wan Shi Tong. Seribu tahun kemudian, Anda yang notabene adalah mahasiswa STEI ITB mencoba mengunjungi perpustakaan tersebut untuk membuktikan apakah Wan Shi Tong benar-benar ada dan masih hidup.

Ternyata, Wan Shi Tong benar-benar ada! Tapi wujudnya seperti burung hantu robot dengan style cyberpunk!



Wan Shi Tong

Source: <https://id.pinterest.com/pin/548102217148927824/?lp=true>

Dia pun bercerita kepada Anda, bahwa perpustakaan tuanya kini semakin sedikit pengunjungnya dan hal itu membuatnya sedih. Anda sebagai calon programmer memutuskan untuk membantu Wan Shi Tong dengan membuatkan sistem manajemen perpustakaan dengan bahasa Pascal. Bisakah Anda membantunya?

Spesifikasi Program

F01 - Registrasi akun

Daftar akun hanya dapat dilakukan oleh admin. Dapat diasumsikan username yang dimasukan tidak pernah dipakai sebelumnya.

\$ register

Masukkan nama pengunjung: **Wan Shi Tong**

Masukkan alamat pengunjung: **Jl. Perpustakaan, Tengah Gurun 40135**

Masukkan username pengunjung: **wshitong997**

Masukkan password pengunjung: **wanshitongganteng**

Pengunjung Wan Shi Tong berhasil terdaftar sebagai user.

F02 - Login

Login dapat dilakukan baik oleh pengunjung maupun oleh admin. Login hanya dapat dilakukan jika pengguna belum login.

\$ login

Masukkan username: **wshitong997**

Masukkan password: **wanshitongganteng**

Selamat datang Wan Shi Tong!

Bila login tidak berhasil (username tidak ditemukan atau password tidak sesuai), pesan kesalahan akan muncul.

\$ login

Masukkan username: **wshitong997**

Masukkan password: **wanshitonggantengz**

Username / password salah! Silakan coba lagi.

F03 - Pencarian buku berdasarkan kategori

Pengunjung dan admin dapat melakukan pencarian buku berdasarkan kategori. Terdapat lima kategori buku **sastra**, **sains**, **manga**, **sejarah**, dan **programming**. Hasil pencarian yang ditampilkan adalah **ID buku**, **judul buku**, dan **penulis buku** yang **diurutkan** berdasarkan **judul** secara leksikografis (pengurutan seperti kamus).

Format output: ID Buku | Judul Buku | Penulis Buku

```
$ cari
Masukkan kategori: programming
```

```
Hasil pencarian:
2516 | Clean Code | Robert C. Martin
786727 | Competitive Programming 3 | Steven Halim
98311 | Javascript for Babies | Sterling Children's
6573 | Learning Java | Jonathan Knudsen
```

Kategori yang dimasukkan belum tentu valid, jadi lakukan input hingga kategori valid (merupakan salah satu dari 5 kategori). Selain itu, mungkin saja tidak ada buku pada kategori tersebut. Apabila tidak ada buku dalam kategori tersebut akan menampilkan pesan tidak ada buku.

```
$ cari
Masukkan kategori: fiksi
Kategori fiksi tidak valid.
Masukkan kategori: Programming
Kategori Programming tidak valid.
Masukkan kategori: programming

Hasil pencarian:
Tidak ada buku dalam kategori ini.
```

F04 - Pencarian buku berdasarkan tahun terbit

Pengunjung dan admin dapat melakukan pencarian buku berdasarkan tahun terbit. Pengguna memasukkan sebuah tahun, misalnya yy, dan kategori pencarian yaitu: { =, <, >, >=, <= }. = artinya buku yang terbit pada tahun yy; < artinya buku yang terbit setelah tahun yy; > artinya buku yang terbit sebelum tahun yy; >= artinya buku yang terbit pada atau setelah tahun yy; <= artinya buku yang terbit sebelum atau pada tahun yy.

Format output: ID Buku | Judul Buku | Penulis Buku

```
$ caritahunterbit
Masukkan tahun: 2001
Masukkan kategori: <

Buku yang terbit < 2001:
2516 | Clean Code | Robert C. Martin
786727 | Competitive Programming 3 | Steven Halim
```

Jika tidak ada buku yang memenuhi kriteria pencarian, tampilkan pesan "Tidak ada buku yang sesuai".

```
$ caritahunterbit
Masukkan tahun: 2010
Masukkan kategori: >

Buku yang terbit > 2010:
Tidak ada buku dalam kategori ini.
```

F05 - Peminjaman buku

Peminjaman buku hanya dapat dilakukan oleh pengunjung. Untuk melakukan peminjaman buku, pengunjung harus melakukan login terlebih dahulu.

```
$ pinjam_buku
Masukkan id buku yang ingin dipinjam: 2516
Masukkan tanggal hari ini: 26/04/2019
Buku Clean Code berhasil dipinjam!
Tersisa 4 buku Clean Code.
Terima kasih sudah meminjam.
```

Dapat diasumsikan id buku dan tanggal yang dimasukkan valid. Namun, bisa saja buku yang ingin dipinjam sedang habis (*out of stock*).

```
$ pinjam_buku
Masukkan id buku yang ingin dipinjam: 2516
Masukkan tanggal hari ini: 26/04/2019
Buku Clean Code sedang habis!
Coba lain kali.
```

F06 - Pengembalian buku

Pengembalian buku hanya dapat dilakukan oleh **pengunjung yang sudah login**. Untuk melakukan pengembalian buku, pengunjung harus melakukan login terlebih dahulu.

```
$ kembalikan_buku
Masukkan id buku yang dikembalikan: 2516
Data peminjaman:
Username: daspro
Judul buku: Clean Code
Tanggal peminjaman: 22/04/2019
Tanggal batas pengembalian: 29/04/2019

Masukkan tanggal hari ini: 26/04/2019
Terima kasih sudah meminjam.
```

Dapat diasumsikan id buku dan format tanggal yang dimasukkan valid. Namun, bisa saja buku terlambat dikembalikan. Jika terlambat mengembalikan, dapat dimunculkan peringatan keterlambatan.

```
> kembalikan_buku
Masukkan id buku yang dikembalikan: 2516
Data peminjaman:
Username: daspro
Judul buku: Clean Code
Tanggal peminjaman: 22/04/2019
Tanggal batas pengembalian: 29/04/2019

Masukkan tanggal hari ini: 01/05/2019
Anda terlambat mengembalikan buku.
```

F07 - Melaporkan buku hilang

Pengunjung dapat menuliskan laporan mengenai buku yang hilang. Untuk membuat laporan buku yang hilang, dibutuhkan masukan berupa ID buku, judul buku, dan tanggal pelaporan buku hilang.

```
$ lapor_hilang
Masukkan id buku: 1998
Masukkan judul buku: Laskar Pelangi
Masukkan tanggal pelaporan: 26/03/2019

Laporan berhasil diterima.
```

Dapat diasumsikan id buku, judul buku, dan format tanggal pelaporan valid.

F08 - Melihat laporan buku yang hilang

Admin bisa menuliskan command untuk menampilkan laporan buku yang hilang.

Format : ID Buku | Judul Buku | Tanggal Pelaporan

```
$ lihat_laporan
Buku yang hilang :
2220 | Malam yang indah | 01/05/2019
2516 | Clean Code | 28/02/2018
3333 | Akal Sehat | 04/03/2019
```

F09 - Menambahkan buku baru ke sistem

Admin bisa menambahkan buku baru yang dibeli oleh pengurus perpustakaan ke dalam sistem perpustakaan.

```
$ tambah_buku
Masukkan Informasi buku yang ditambahkan:
Masukkan id buku: 2001
Masukkan judul buku: 97 things every programmer should know
Masukkan pengarang buku: Kevlin Henney
Masukkan jumlah buku: 10
Masukkan tahun terbit buku: 2010
Masukkan kategori buku: programming

Buku berhasil ditambahkan ke dalam sistem!
```

Dapat diasumsikan semua masukan valid (id buku unik, kategori buku merupakan salah satu dari 5 kategori, jumlah buku lebih dari 0, dll).

F10 - Melakukan penambahan jumlah buku ke sistem

Admin bisa melakukan pembaharuan jumlah buku yang tersedia ke sistem.

```
$ tambah_jumlah_buku
Masukkan ID Buku: 2001
Masukkan jumlah buku yang ditambahkan: 15

Pembaharuan jumlah buku berhasil dilakukan, total buku 97 things every programmer
should know di perpustakaan menjadi 25
```

Pada contoh di atas, buku "97 things every programmer should know" mulanya hanya 10 buku, kini menjadi 25 buku. Dapat diasumsikan id buku valid (sudah ada di data) dan jumlah buku yang ditambahkan valid (bernilai lebih dari 0).

F11 - Melihat riwayat peminjaman

Admin bisa melihat riwayat dari peminjaman buku seorang pengunjung.

Format : Tanggal_Pengembalian | ID_Buku | Judul Buku

```
$ riwayat
Masukkan username pengunjung: wshitong997
Riwayat:
15/03/2019 | 2247 | Motivasi Burger King
29/03/2019 | 2001 | 97 things every programmer should know
14/04/2019 | 3753 | Laskar Pelangi
```

Dapat diasumsikan username valid (sudah terdaftar pada data).

F12 – Statistik

Admin bisa melihat statistik yang berkaitan dengan pengguna dan buku. Statistik pengguna berisi daftar banyak pengguna per kategori admin dan pengunjung. Statistik buku berisi jumlah total buku pada tiap kategori.

Berikut contoh statistik:

```
$ statistik
Pengguna:
Admin | 2
Pengunjung | 4
Total | 6
Buku:
sastra | 40
sains | 21
manga | 42
sejarah | 0
programming | 53
Total | 156
```

F13 - Load file

Program pertama kali akan menjalankan fungsi load.

```
$ load
Masukkan nama File Buku: buku.csv
Masukkan nama File User: user.csv
Masukkan nama File Peminjaman: peminjaman.csv
Masukkan nama File Pengembalian: pengembalian.csv
Masukkan nama File Buku Hilang: kehilangan.csv

File perpustakaan berhasil dimuat!
```

Dapat diasumsikan nama file valid dan struktur data file sesuai definisi di bawah.

F14 - Save file

User bisa menjalankan penyimpanan data dengan menjalankan fungsi save file. File disimpan sesuai definisi struktur data di bawah.

```
$ save
Masukkan nama File Buku: buku.csv
Masukkan nama File User: user.csv
Masukkan nama File Peminjaman: peminjaman.csv
Masukkan nama File Pengembalian: pengembalian.csv
Masukkan nama File Buku Hilang: kehilangan.csv

Data berhasil disimpan!
```

F15 - Pencarian anggota

Admin bisa menjalankan fungsi untuk mencari data diri dari anggota perpustakaan. Jika username tidak ditemukan, tampilkan pesan kalau anggota tidak ditemukan

```
$ cari_anggota
Masukkan username: wshitong997
Nama Anggota: Wan Shi Tong
Alamat anggota: Jl. Perpustakaan, Tengah Gurun 40135
```

F16 - Exit

User dapat menjalankan perintah exit. Saat perintah ini dijalankan, user akan ditanya apakah mereka akan menyimpan data.

```
$ exit
Apakah anda mau melakukan penyimpanan file yang sudah dilakukan (Y/N) ? Y
```

Spesifikasi Bonus

B01 - Penyimpanan Password

Dalam *software engineering*, *password* tidak pernah disimpan secara polos, melainkan di-*hash*. Hal ini dilakukan supaya *password* pengguna tetap aman meskipun file csv dicuri orang lain. Karena itu, Anda diminta mengenkripsi atau melakukan *hash* pada *password* sebelum *password* disimpan ke file csv. Algoritma enkripsi atau *hash* dibebaskan kepada mahasiswa. Semakin baik algoritma yang digunakan, poin yang didapat akan semakin tinggi.

B02 - Denda

Pada saat pengunjung mengembalikan buku (fungsionalitas F05) namun tanggal hari ini lebih dari tanggal pengembalian, selain menampilkan pesan peringatan terlambat mengembalikan, program juga harus menuliskan denda yang perlu dibayarkan. Denda yang dikenakan adalah Rp 2.000,00 tiap harinya. Contoh kasus pengembalian buku yang terkena denda:

```
> kembalikan_buku
Masukkan id buku yang dikembalikan: 2516
Data peminjaman:
Username: daspro
Judul buku: Clean Code
Tanggal peminjaman: 22/04/2019
Tanggal batas pengembalian: 29/04/2019

Masukkan tanggal hari ini: 01/05/2019
Anda terlambat mengembalikan buku.
Anda terkena denda 4000.
```

Catatan Tambahan

1. Pembagian fitur atau fungsionalitas di atas tidak merepresentasikan dekomposisi modul, fungsi, dan prosedur yang sesungguhnya.
2. Semua masukan program, kecuali disebutkan secara khusus, tidak perlu dilakukan validasi.

Struktur Data File Eksternal

Program perlu membaca beberapa data dari file eksternal untuk mengoperasikan sistem ini. Format file eksternal yang diminta adalah file dengan ekstensi **.csv**

File Buku

ID_Buku, Judul_Buku, Author, Jumlah_Buku, Tahun_Penerbit, Kategori

Keterangan:

- ID_Buku: nomor identitas buku, identitas haruslah berupa numerik
- Judul_Buku: judul dari buku
- Author: pengarang buku
- Jumlah_Buku: jumlah buku yang ada di perpustakaan
- Tahun_Penerbit: Tahun diterbitkannya buku oleh penerbit
- Kategori : Kategori buku, sastra/sains/manga/sejarah/programming

File User

Nama, Alamat, Username, Password, Role

Keterangan:

- Nama: nama pengguna
- Alamat: alamat tempat tinggal pengguna
- Username: username pengguna
- Password: password pengguna
- Role: role pengguna (Pengunjung / Admin) untuk menentukan akses masuk pengguna.

File History Peminjaman

Username, ID_Buku, Tanggal_Peminjaman, Tanggal_Batas_Pengembalian, Status_Pengembalian

Keterangan:

- Username: username akun yang meminjam buku
- ID_Buku: nomor ID buku yang dipinjam, identitas haruslah berupa numerik
- Tanggal_Peminjaman: Tanggal peminjaman buku
- Tanggal_Batas_Pengembalian: Tanggal batas pengembalian buku
- Status_Pengembalian: Status pengembalian buku (sudah/belum. Boleh dituliskan dalam boolean)

File History Pengembalian

Username, ID_Buku, Tanggal_Pengembalian

Keterangan:

- Username: username yang mengembalikan buku
- ID_Buku: nomor ID buku yang dipinjam, identitas haruslah berupa numerik
- Tanggal_Pengembalian: tanggal pengembalian buku

File Laporan Buku Hilang

Username, ID_Buku_Hilang, Tanggal_Laporan

Keterangan:

- Username: Username pelapor buku
- ID_Buku_Hilang: nomor ID buku yang hilang, identitas haruslah berupa numerik
- Tanggal_Laporan: Tanggal Pelaporan

Format Tanggal

Untuk penulisan tanggal, gunakan format DD/MM/YYYY.

Keterangan:

- DD: Tanggal direpresentasikan dalam 2 angka (contoh: 15)
- MM: Bulan direpresentasikan dalam 2 angka (contoh: Maret menjadi 03)
- YYYY: Tahun direpresentasikan dalam 4 angka (contoh: 2019)

Deliverables

1. *Source code* program yang dibuat sesuai standar yang diajarkan di kuliah, bersih (hanya mengandung bagian-bagian yang diperlukan), dan *well-commented*.
2. **Laporan Tugas Besar** dibuat dengan template sebagai berikut:
 - a. Halaman Cover, berisi minimum Kode dan Nama Kuliah, Nama Tugas, Nomor Kelas, Nomor Kelompok, NIM dan Nama Anggota Kelompok, Nama Sekolah (Sekolah Teknik Elektro dan Informatika), dan Perguruan Tinggi (Institut Teknologi Bandung), dan Tahun (2019).
 - b. Daftar Isi.
 - c. Daftar Tabel (jika ada).
 - d. Daftar Gambar (jika ada).
 - e. Deskripsi persoalan: berisi penjelasan kembali (dengan "bahasa sendiri") tentang persoalan yang akan diselesaikan.
 - f. Daftar pembagian kerja anggota kelompok, berdasarkan fitur yang dibuat, meliputi desain, implementasi/koding, dan testing.

Contoh:

Fitur	Implementasi *)	NIM Desainer **)	NIM Coder **)	NIM Tester **)
F01-Register akun	procedure daftar procedure tambah_user	16518501 16518511	16518511	16518524
F02-Login akun	procedure login	16518511	16518524	16518511
F03-Pencarian buku	procedure cari_buku	16518524	16518501	16518501
...

*) bisa menjadi procedure/fungsi/bagian dari program utama dan bisa lebih dari 1

**) bisa dikerjakan oleh lebih dari 1 orang

- g. Checklist hasil rancangan, implementasi dan testing setiap primitif.

Contoh Checklist:

Fitur	Desain	Implementasi	Testing
F01-Register akun	V	X	X
F02-Login akun	V	V	V
F03-Pencarian buku	X	-	-
...

Keterangan: V: sudah selesai dikerjakan, X: dikerjakan, tapi belum selesai, -: tidak dikerjakan sama sekali.

- h. Desain command untuk setiap primitif (berisi: nama command, masukan, dan keluaran).
- i. Desain kamus data (dalam notasi algoritmik).
- j. Desain dekomposisi algoritmik dan fungsional program.
- k. Spesifikasi untuk tiap modul/prosedur/fungsi yang dibuat (dalam notasi algoritmik).
- l. Berikan *screenshot* hasil pengujian program berdasarkan fitur-fitur pada spesifikasi. Pada setiap fitur, *screenshot* minimal berisi: (1) data masukan, (2) data keluaran
- m. Lampiran: Hasil scan form asistensi.