

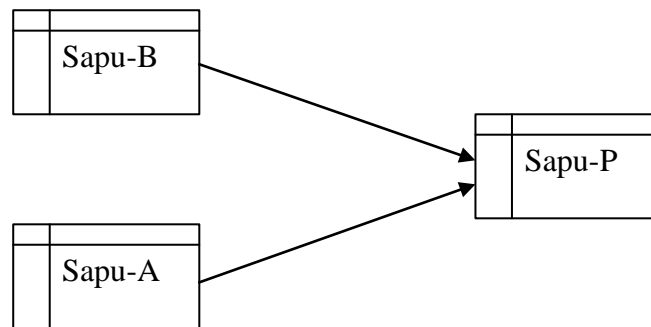
# Spesifikasi Tugas Besar Dasar Pemrograman 2013/2014

## Sistem Administrasi Perpustakaan (SAPU)

### Deskripsi Produk:

SAPU merupakan aplikasi untuk memproses peminjaman buku pada perpustakaan  
SAPU terdiri atas 3 modul yaitu:

1. SAPU-B : menangani administrasi data referensi buku pada perpustakaan
2. SAPU-A : menangani administrasi data master anggota perpustakaan
3. SAPU-P : menangani administrasi data transaksi peminjaman buku



### SAPU-B

Modul SAPU-Buku menggunakan struktur data sebagai berikut.

```
type Buku = <string, string, string, string, integer>0  
{type Buku terdiri atas kodeBuku:string, JudulBuku:string, Pengarang:String,  
Penerbit:String, TahunTerbit:integer>0 }
```

Primitif yang diperlukan sebagai berikut:

1. CreateBuku : string, string, string, string, integer >0 → Buku  
{ CreateBuku(K, J,PR,PB, T) membentuk sebuah objek Buku }
2. GetKodeBuku : Buku → string  
{ GetKodeBuku(B) mengakses atribut kodeBuku dari B }
3. GetJudulBuku : Buku → string  
{ GetJudulBuku(B) mengakses atribut JudulBuku dari B }
4. GetPengarangBuku : Buku → string  
{ GetPengarangBuku(B) mengakses atribut PengarangBuku dari B }
5. GetPenerbitBuku : Buku → string  
{ GetPenerbitBuku(B) mengakses atribut PenerbitBuku dari B }
6. GetTahunTerbitBuku : Buku → Integer>0  
{ GetTahunTerbitBuku(B) mengakses atribut TahunTerbitBuku dari B }
7. CopyBuku : Buku → Buku  
{ CopyBuku(B) menghasilkan objek Buku baru salinan dari B }
8. KonsoListBuku : Buku , list of Buku → list of Buku  
{ KonsoListBuku(B,L) menyisipkan B sebagai elemen pertama list B }
9. IsMemberListBuku : Buku , list of Buku → boolean  
{ IsMemberListBuku(B,L) memeriksa apakah B adalah anggota list L }
10. NbElemenListBuku : list of Buku → integer  
{ NbElemenListBuku(L) menghitung banyak elemen list L }
11. PrintBuku : Buku → nil  
{ PrintBuku(B) mencetak isi Buku ke layar }
12. PrintListBuku : list of Buku → nil  
{ PrintListBuku(L) mencetak isi list L ke layar }

## SAPU-A

Modul SAPU-A menggunakan struktur data sebagai berikut.

```
type Anggota = <string, string, string>
{type Anggota terdiri atas kodeAnggota:string, nama:string, Alamat: string }
```

Primitif yang diperlukan sebagai berikut:

1. CreateAnggota : string, string, string → Anggota  
{CreateAnggota(K, N, A) membentuk sebuah objek Anggota}
2. GetKode : Anggota → string  
{GetKode(A) mengakses atribut KodeAnggota dari A}
3. GetNama : Anggota → string  
{GetNama(A) mengakses atribut Nama dari A}
4. GetAlamat : Anggota → string  
{GetAlamat(A) mengakses atribut Alamat dari A}
5. CopyAnggota : Anggota → Anggota  
{CopyAnggota(A) menghasilkan objek Anggota baru salinan dari A}
6. KonsoAnggota : Anggota , list of Anggota → list of Anggota  
{KonsoAnggota(A,L) menyisipkan A sebagai elemen pertama list L}
7. IsMemberAnggota : Anggota , list of Anggota → boolean  
{IsMemberAnggota(A,L) memeriksa apakah A adalah anggota list L}
8. NbElemenAnggota : list of Anggota → integer  
{NbElemenAnggota(L) menghitung banyak elemen list L}
9. PrintAnggota : Anggota → nil  
{PrintAnggota(A) mencetak isi Anggota ke layar}
10. PrintListAnggota : list of Anggota → nil  
{PrintListAnggota(L) mencetak isi list L ke layar}

## SAPU-P

Modul SAPU-P menggunakan struktur data dalam modul SAPU-B, SAPU-A, dan kamus sebagai berikut.

```
type Pinjam = <string, string, string, date, integer > 0 >
{type Pinjam terdiri atas kodePinjam:string, KodeBuku:string,
kodeAnggota:string, TglPinjam:date, Jumlah:integer> 0 }
```

Primitif yang diperlukan sebagai berikut:

1. CreatePinjam : string, string, string, date, integer > 0, list of Buku  
, list of Anggota → Pinjam  
{CreatePinjam(KP, KB, KA, TP, J, LB,LA) membentuk sebuah objek Pinjam dengan syarat KB terdaftar di LB, KA terdaftar di LA}
2. GetKodePinjam : Pinjam → string  
{GetKodePinjam(P) mengakses atribut kodePinjam dari P}
3. GetKodeBuku : Pinjam → string  
{GetKodeBuku(P) mengakses atribut kodeBuku dari P}
4. GetKodeAnggota : Pinjam → string  
{GetKodeAnggota(P) mengakses atribut kodeAnggota dari P}
5. GetTglPinjam : Pinjam → string  
{ GetTglPinjam (P) mengakses atribut TglPinjam dari P}
6. GetJumlah : Pinjam → string  
{GetJumlah(P) mengakses atribut Jumlah dari P}
7. CopyPinjam : Pinjam → Pinjam  
{CopyPinjam(P) menghasilkan objek Pinjam baru salinan dari P}
8. KonsoListPinjam : Pinjam , list of Pinjam → list of Pinjam  
{ KonsoListPinjam (P,L) menyisipkan P sebagai elemen pertama list L}

9. IsMemberListPinjam : Pinjam , list of Pinjam  $\rightarrow$  boolean  
{IsMemberListPinjam(P,L) memeriksa apakah P adalah anggota list L}
10. NbElemenListPinjam : list of Pinjam  $\rightarrow$  integer  
{NbElemenListPinjam(L) menghitung banyak elemen list L}
11. PrintPinjam : Pinjam  $\rightarrow$  nil  
{PrintPinjam(P) mencetak isi Pinjam ke layar}
12. PrintListPinjam : list of Pinjam  $\rightarrow$  nil  
{PrintListPinjam(L) mencetak isi list L ke layar}

### **Deskripsi Tugas:**

- Kelompok B1 dan B2 mengerjakan modul SAPU-B
- Kelompok A1 dan A2 mengerjakan modul SAPU-A
- Kelompok P1 dan P2 mengerjakan modul SAPU-P
- untuk kemudian modul B,A,P tersebut dapat diintegrasikan menjadi satu kesatuan aplikasi SAPU.
- Terdapat 3 grup besar developer SAPU yaitu grup Sapu1 (Kelompok B1, A1, dan P1) dan Sapu2 (Kelompok B2, A2, dan P2)
- Jika diperlukan, tambahkan fungsi/prosedur lain serta asumsi yang relevan.

### **Deliverables:**

1. Tugas dikumpulkan perkelompok maksimum tanggal 30 Desember 2013 ke [sutikno.wae@gmail.com](mailto:sutikno.wae@gmail.com) dan [helmie.arif@gmail.com](mailto:helmie.arif@gmail.com)
2. File yang dikumpulkan berupa file terkompresi (TBDasPro-XX.zip/rar) sesuai kelompok {contoh: TBDaspro-B2 untuk kelompok B2}. File tersebut terdiri atas 2 direktori : SOURCE{berisi file source code LISP dan file data} dan DOK{berisi file laporan pengerjaan tugas termasuk strategi yang digunakan}