**void insertGuru(List\_Guru &L, adr\_Guru P);**

IS: Terdefinisi List Guru & Pointer yg menunjuk ke Elm\_Guru  
FS: Elm\_Guru yang ditunjuk oleh P menjadi L.first dari List\_Guru

**void insertMurid(List\_Murid &L, adr\_Murid P);**

IS: Terdefinisi List Murid & Pointer yg menunjuk ke Elm\_Murid

FS: Elm\_Murid yang ditunjuk oleh P menjadi L.last dari List\_Murid

**void insertRelasi(List\_relasi &L, adr\_relasi P);**

IS: Terdefinisi List Relasi & Pointer yg menunjuk ke Elm\_Relasi

FS: Elm\_ Relasi yang ditunjuk oleh P menjadi L.first dari List\_ Relasi

**void deleteGuru(List\_Guru &L, adr\_Guru &P);**

IS: Terdefinisi List Guru & Pointer yg menunjuk NULL

FS: Elm\_Guru pertama ditunjuk oleh P dan dihapus dari List\_Guru

**void deleteMurid(List\_Murid &L, adr\_Murid &P);**

IS: Terdefinisi List Murid & Pointer yg menunjuk NULL

FS: Elm\_Murid Terakhir ditunjuk oleh P dan dihapus dari List\_Murid

**void deleteRelasi(List\_relasi &L, adr\_relasi P);**

IS: Terdefinisi List Relasi & Pointer yg menunjuk NULL

FS: Elm\_Relasi pertama ditunjuk oleh P dan dihapus dari List\_Relasi

**void checkRelasi(List\_relasi L, adr\_Murid P, adr\_Guru Q);**

IS: : Terdefinisi List Relasi & Pointer P yg menunjuk Elm\_Murid serta & Pointer Q yg menunjuk Elm\_Guru

FS: Mengecek apakah kedua Elemen yang ditunjuk memiliki relasi

**void showGuru(List\_Guru L);**

IS: Terdefinisi List Guru

FS: Menampilkan semua elemen pada List Guru

**void showMurid(List\_Murid L);**

IS: Terdefinisi List Murid

FS: Menampilkan semua elemen pada List Murid

**void showMurid\_dariGuru(List\_relasi L, adr\_Guru P);**

IS: Terdefinisi List Relasi dan pointer P yang menunjuk Elm\_Guru

FS: Menampilkan semua info dari Elm\_Murid yang memiliki relasi dengan P

**void showGuru\_dariMurid(List\_relasi L, adr\_Murid P);**

IS: Terdefinisi List Relasi dan pointer P yang menunjuk Elm\_Murid

FS: Menampilkan semua info dari Elm\_Guru yang memiliki relasi dengan P

**void showAll\_relasiGuru(List\_Guru Ld, List\_relasi Lr);**

IS: Terdefinisi List Guru dan List Relasi

FS: Menampilkan relasi dari setiap Elm\_Guru pada List\_Guru

**void showAll\_relasiMurid(List\_Guru Lm, List\_relasi Lr);**

IS: Terdefinisi List Murid dan List Relasi

FS: Menampilkan relasi dari setiap Elm\_Murid pada List\_ Murid

**void countMurid(List\_relasi L, adr\_Guru P);**

IS: Terdefinisi List Relasi dan sebuah pointer P yang menunjuk pada Elm\_Guru

FS: Menampilkan jumlah Elm\_Murid yang memiliki relasi dengan P

**void countGuru(List\_relasi L, adr\_Murid P);**

IS: Terdefinisi List Relasi dan sebuah pointer P yang menunjuk pada Elm\_Murid

FS: Menampilkan jumlah Elm\_Dose yang memiliki relasi dengan P

**void countMurid\_Less(List\_relasi Lr, List\_Guru Ld);**

IS: Terdefinisi List Relasi dan List Guru

FS: Menampilkan jumlah Elm\_Murid yang tidak memiliki Relasi dengan Elm\_Guru

**void countGuru\_Less(List\_relasi Lr, List\_Murid Lm);**

IS: Terdefinisi List Relasi dan List Murid

FS: Menampilkan jumlah Elm\_Guru yang tidak memiliki Relasi dengan Elm\_Murid

**void editGuru(List\_relasi L, adr\_Guru P, adr\_Murid Q, adr\_Murid R);**

IS: Terdefinisi List Relasi dan 3 Pointer yang menunjuk 1 Elm\_Guru & 2 Pointer yang menunjuk Elm\_Murid, pointer P dan Q memiliki relasi

FS: Relasi dari pointer Q dengan P diganti menjadi relasi pointer R

**void editMurid(List\_relasi L, adr\_Murid P, adr\_Guru Q, adr\_Guru R);**

IS: IS: Terdefinisi List Relasi dan 3 Pointer yang menunjuk 1 Elm\_Murid & 2 Pointer yang menunjuk Elm\_Guru, pointer P dan Q memiliki relasi

FS: Relasi dari pointer Q dengan P diganti menjadi relasi pointer R