

9. STRUKTUR, UNION & PENOMBORAN

9.0 Pengenalan

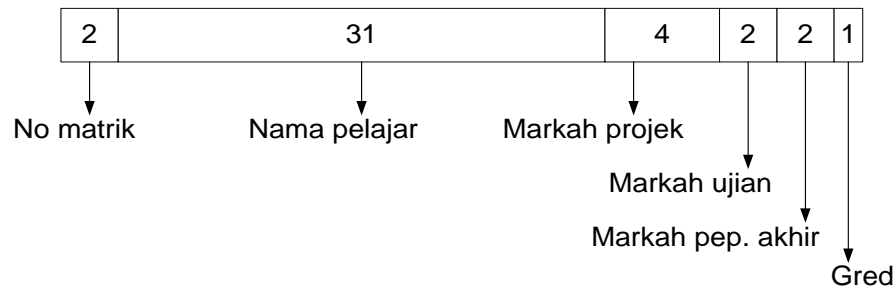
- **Struktur** -
 - satu kumpulan pembolehubah yg dikumpulkan & dirujuk di bawah satu nama @ pencam.
 - jenis data yang ditakrif oleh pengguna
 - data terdiri dari jenis yang berlainan yang membentuk maklumat mengenai sesuatu data
 - berbeza dengan tatasusunan yang terdiri dari jenis data yang sama
- **Pengisytiharan:**

```
struct <nama_struktur>
{
    <jenis_data> <nama_unsur_1>;
    <jenis_data> <nama_unsur_2>;
    :
    <jenis_data> <nama_unsur_n>;
} <senarai_p'ubah_struktur>;
```

- **Cth:**
Seorang pensyarah hendak menentukan gred kepada setiap pelajar-pelajarnya pada setiap semester. Pelajar-pelajar telah menjalani projek, ujian dan peperiksaan akhir.

<i>Nama Data</i>	<i>Jenis Data</i>	<i>C</i>
No matrik	Nilai integer	int
Nama pelajar	Tatasusunan aksara 31	char [31]
Markah projek	Nilai titik terapung	float
Markah ujian	Nilai integer	int
Markah pep. akhir	Nilai integer	int
Gred	Aksara	char

- Saiz ingatan yg diperlukan utk struktur ini dlm bait:



```

struct
{
    int no_metrik;
    char    nama[31];
    float   markah_projek;
    int markah_ujian;
    int markah_final;
    char gred;
} rekod_pelajar;

struct pelajar
{
    int no_metrik;
    char    nama[31];
    float   markah_projek;
    int markah_ujian;
    int markah_final;
    char gred;
} rekod_pelajar;

struct pelajar
{
    int no_metrik;
    char    nama[31];
    float   markah_projek;
    int markah_ujian;
    int markah_final;
    char gred;
};
struct pelajar rekod_pelajar;

```

Cth pengisytiharan i

Cth pengisytiharan ii

Cth pengisytiharan iii

- Satu lagi pilihan pengisytiharan struktur menggunakan typedef, cth:

```
typedef struct
{
    int no_metrik;
    char    nama[31];
    int markah_projek1;
    int markah_projek2;
    int markah_projek3;
    int markah_ujian1;
    int markah_ujian2;
    int markah_final;
    char gred;
} JENISPELAJAR;

JENISPELAJAR rekod_pelajar;
```

9.1 Umpukan nilai awal kpd struktur

```
struct pelajar
{
    int    no_metrik;
    char    nama[31];
    float    markah_projek;
    int    markah_ujian;
    int    markah_final;
    char gred;
} rekod_pelajar = {722, "Maimunah Muhamad", 30.5,
10, 50, 'B'};
```

Atau

```

struct pelajar
{
    int no_metrik;
    char    nama[31];
    float markah_projek;
    int markah_ujian;
    int markah_final;
    char gred;
};
struct pelajar rekod_pelajar = {722, "Maimunah
Muhamad", 30.5, 10, 50, 'B'};

```

9.2 Operasi ke atas unsur struktur

- Setelah diisytiharkan, unsur struktur boleh dirujuk spt berikut:

```
<p'ubah_struktur>.<nama_unsur>
```

- Contoh cetakan nama pelajar dibuat:

```
printf("%s", rekod_pelajar.nama);
```

- Contoh mengumpukan nilai ke no_metrik:

```
rekod_pelajar.no_metrik = 5443;
```

- Untuk merujuk elemen tertentu didlm tatasusunan *subskrip* @ *indek* perlu didunakan. Cth menyalin nama pelajar:

```

struct pelajar p1, p11;
for (i=0; i<31; i++)
    p11.nama[i] = p1.nama[i];

```

Latihan

Tulis satu aturcara yang menggunakan struktur utk bagi tujuan penempahan tiket kapal terbang. Minimum data yang perlu ada adalah : Nama penumpang, tempat berlepas, destinasi, no. penerbangan, jumlah yg. perlu dibayar, status bayaran [telah dibayar|belum dibayar] dan umur penumpang.

9.2 Struktur Bersarang

- ahli struktur boleh terdiri dari struct yg lain.
- Perhatikan pengisytiharan struktur berikut:

```

struct struktur_pelajar
{
    char    nama[31];
    int no_metrik;
    char kod_fakulti[5];
    int umur;
    char jantina;
    int markah_projek1;
    int markah_projek2;
    int markah_projek3;
    int markah_ujian1;
    int markah_ujian2;
    int markah_final;
    char gred;
} rekod_pelajar;

```

- menstrukturkan semula rekod pelajar menggunakan pengisytiharan supaya data boleh diklasifikasi mengikut kategorinya:
 - peribadi pelajar
 - markah pelajar

Cara 1:

```
struct peribadi_pelajar
{
    char    nama[31];
    int no_metrik;
    char kod_fakulti[5];
    int umur;
    char jantina;
}

struct markah_pelajar
{
    int markah_projek1;
    int markah_projek2;
    int markah_projek3;
    int markah_ujian1;
    int markah_ujian2;
    int markah_final;
}

struct pelajar
{
    struct peribadi_pelajar  peribadi;
    struct markah_pelajar markah;
    char gred;
};
```

Cara 2:

```

struct pelajar
{
    struct
    {
        char    nama[31];
        int no_metrik;
        char kod_fakulti[5];
        int umur;
        char jantina;
    } peribadi;

    struct
    {
        int markah_projek1;
        int markah_projek2;
        int markah_projek3;
        int markah_ujian1;
        int markah_ujian2;
        int markah_final;
    } markah;

    char gred;
};

```

- Jika 2 p'ubah diisytiharkan:

```
struct pelajar p1, p2;
```

Kenyatan² berikut ada adalah sah:

```

p1.peribadi.umur = p2.peribadi.umur;
p1=p2;
p1.peribadi.kod_fakulti[4] = p2.peribadi.nama[9];
strcpy(p1.peribadi.nama,"Amirah Iman");
p1.gred = 'A';

```

9.2.1 Umpukan nilai awal kpd struktur

- Cth:

```
struct pelajar p1 = {{“Maimunah Muhamad”, 722,
    FS, 20, P},{10, 10, 20,0,12,60}, ‘C’};
```

Latihan

Dari latihan 1.3, tambahkan fungsi berikut. Setiap penempah boleh menempah lebih dari 1 tiket utk penerbangan yg sama, ttp utk orang yg berbeza. Dengan menggunakan struktur bersarang tambahkan data: no. telepon penempah, nama penempah, tarikh & masa tempahan dibuat.

9.3 P'ubah struktur jenis tatasusunan

- Penisytiharan:

```
struct <n_struk> <n_tts>[dms];
```

- cth pengisytiharan:

```
struct pelajar
{
    int no_metrik;
    char nama[31];
    float markah_projek;
    int markah_ujian;
    int markah_final;
    chargred;
};
struct pelajar rekod_pelajar[3];
```


- Saiz tts boleh diabaikan jika nilai awal diber, cth:

```
struct pelajar rekod_pelajar[] = {
    {2332, "Zainab Ismail", 10.2, 10, 20,
    'F'},
    {2335, "Ali Mamat", 51.4, 60, 60, 'C'},
    {2337, "Malia Azri", 100.0, 99, 99, 'A'}
};
```

- Tts yg terbentuk:

no_metrik	nama	markah_projek	markah_ujian	markah_final	gred
2332	Zainab Ismail	10.2	10	20	F
2335	Ali Mamat	51.4	60	60	C
2337	Malia Azri	100.0	99	99	A

- Cth memasukkan maklumat kedlm tts struktur:

```

struct pensyarah_fsksm
{
    int no_pekerja;
    struct
    {
        char nama_awal[15];
        char nama_akhir[15];
    } nama;
    int umur;
    float gaji;
} pensy[100];

```

```

for (j=0; j<100; j++)
{
    printf("\nmasukkan no pek, nama awal, nama
        akhir umur dan gaji %d\n",j+1);
    scanf("%d", pensy[j].no_pekerja);
    gets(pensy[j].nama.nama_awal);
    gets(pensy[j].nama.nama_akhir);
    scanf("%d", pensy[j].umur);
    scanf("%f",pensy[j].gaji);
}

```

Latihan

Dari latihan 2.2, tambahkan fungsi berikut. Boleh menyimpan 20 data dlm tts untuk tujuan:

- kiraan bil penumpang dlm pesawat tertentu
- jumlah tambang yg telah dibayar & belum dibayar.

9.4 P'ubah struktur jenis Penuding

- Pengisytiharan:

```
struct <n_struk> *<n_penuding>;
```

- Perhatikan cth berikut:

```
struct pensyarah_fsksm
{
    int no_pekerja;
    struct
    {
        char nama_awal[15];
        char nama_akhir[15];
    } nama;
    int umur;
    float gaji;
}dayang = {777, {"Dayang", "Jawawi"}, 30, 1500};
struct pensyarah_fsksm *saya, *kami;
```

- Cth umpukan:

```
saya = &dayang;
```

- Cth merujuk ahli struktur menggunakan penuding:

```
(*saya).no_pekerja
@
saya->no_pekerja
```

- Sambungan cth di atas:

```

struct pensyarah_fsksm pblk162[]= {
    {777, {"Dayang", "Jawawi"}, 20, 1500},
    {666, {"Normal", "Jusoh"}, 30, 2500}
}

printf("Alamat #1: %u #2: %u\n", &pblk162[0],
      &pblk162[1]);

kami = &pblk162[0];

printf("Penuding #1: %u #2: %u\n", kami, kami+1);

printf("1. Kami adalah %s:", kami->nama.nama_awal);

kami++;

printf("2. Kami adalah %s:", kami->nama.nama_awal);

```

- Lengkapi rajah berikut & tulis output dari cth aturcara di atas.

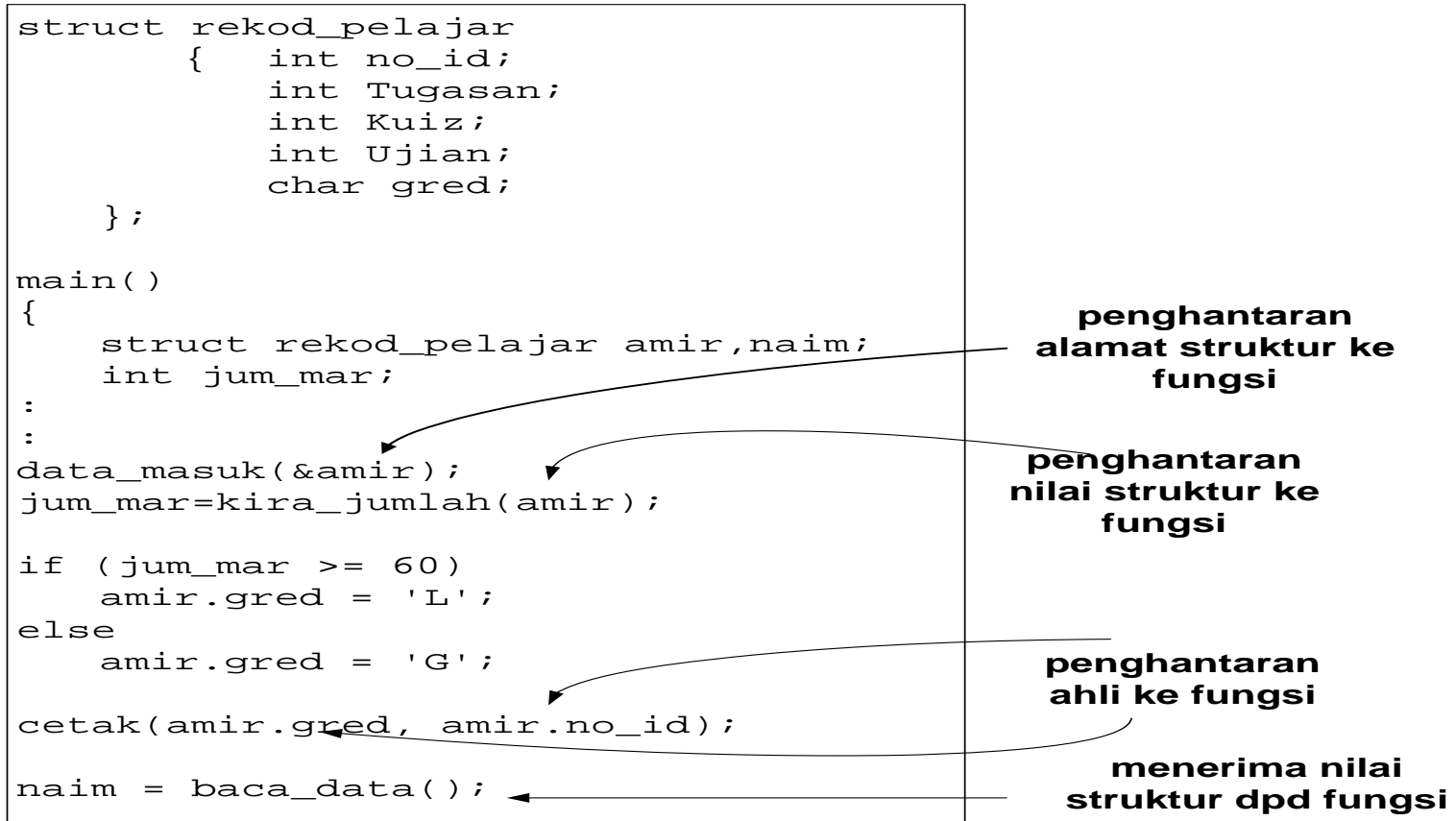
kami

pblk162[0]	777	Dayang 1500
pblk162[0]	666	Normal. 2500

Latihan

Dari latihan 3.1, tukar tts kepada penuding.

9.5 Operasi² struktur dgn fungsi



```
void data_masuk(struct rekod_pelajar *p)
{
    scanf("%d", &(*p).no_id);
    scanf("%d%d",&p->Tugasan, &p->Kuiz);
    scanf("%d", &p->Ujian);
}

int kira_jumlah(struct rekod_pelajar p)
{
    int jum;
    jum = p.Tugasan + p.Kuiz + p.Ujian;
    return jum;
}

void cetak(char gred, int id)
{
    printf("Gred pelajar %d ialah %c",
        id, gred);
}

struct rekod_pelajar baca_data()
{
    struct rekod_pelajar p;
    scanf("%d", &p.no_id);
    scanf("%d%d",&p.Tugasan, &p.Kuiz);
    scanf("%d", &p.Ujian);
    return p;
}
```

menerima struktur
melalui alamat

menerima struktur
melalui nilai

menerima ahli

pengantaran nilai ke
fungsi main

Latihan

Dari latihan 3.1@ 4.1, anda dikehendaki membentukan sekurang-kurangnya 3 fungsi berikut:

- baca_masukan
- pengiraan
- cetak_laporan

9.6 Struktur: Apa seterusnya?

- Jelas struktur amat berguna utk membentuk data jenis baru yg terdiri dpd jenis² data asas.
- Bentuk data yg berbagai jenis ini mewakili data sebenar dlm. penyelesaian masalah dgn komputer.
- Bentuk data ini akan distukturkan mengikut struktur-struktur tertentu spt tindanan, baris gilir & pehohon sebelum diproses.

9.7 Union

- berfungsi lebih kurang sama seperti struktur
- diguna untuk menyimpan jenis data yang berlainan didalam lokasi memori yang sama
- boleh menjimatkan ruang bila pembolehubah tidak digunakan pada satu masa
- hanya satu ahli dari satu jenis data diguna pada satu masa
- pengguna menentukan bila setiap ahli sesuai digunakan

- Cth mudah:

```
#include <stdio.h>
union nombor {
    char x;
    int y;};

main( )
{
    union nombor nilai;
    printf("saiz union ialah %d\n",
        sizeof(nilai));
    nilai.x = 0x41; /*ASCII 'A'*/
    printf("char: %X \t int:  %X\n", nilai.x,
        nilai.y);
    printf("alamat bermula pada  %d\n",
        &nilai.x);
    nilai.y = 0x1234;
    printf("char: %X \t int:  %X\n", nilai.x,
        nilai.y);
    printf("alamat bermula pada  %d\n",
        &nilai.y);
    return 0;
}
```

- Output:

```
saiz union ialah 2
char: 41          int:  41
alamat bermula pada 6692
char: 34          int: 1234
alamat bermula pada 6692
```

- Bila nilai.x diumpukkan:

alamat **nilai**

6692

00	41
----	----

- Bila nilai.y diumpukkan:

alamat **nilai**

6692

12	34
----	----

9.8 Pernomboran

- set nilai-nilai pemalar integer yang diwakili oleh *pecam*.

- Pengisytiharan:

```
enum <nama_jenis_enum>
{
    senarai_enum
};
```

- Cth pengisytiharan jenis data `BulanIslam` & p'ubah:

```
enum BulanIslam
{
    muharram, safar, rabiulawal,
    rabiulakhir, jamadilawal, jamadilakhir,
    rejab, syaaban, ramadhan, syawal,
    zulkaedah, zulhijjah
};
enum BulanIslam Bulan;
```

- Apabila jenis data per nomboran diisytiharkan, *pecam-pecamnya* akan diumpukkan nilai integer secara automatik.
- *Pecam* yang pertama akan diumpukkan nilai '0', *pecam* kedua bernilai '1' dan begitulah seterusnya.
- Pengaturcara boleh membuat pengubahsuaian berdasarkan peraturan-peraturan berikut:
 - boleh mengumpukkan nilai pemalar (*constant*) kepada senarai per nomboran
 - boleh mengumpukkan nilai kepada senarai per nomboran dengan menggunakan *pecam* yang sebelumnya dalam ungkapan aritmetik
 - boleh mengumpukkan nilai yang sama kepada lebih dari satu *pecam* dalam senarai per nomboran

- Cth: Rekabentuk mesin jual air automatik.

```
enum siling
{
    satusen = 1, limasen =5,sepuluhsen= 10,
    samah = limasen*sepuluhsen,serial =
    2*samah
};
enum siling duitmasuk;
```

Untuk kira duit yg dimasukkan oleh pembeli cukup @ tidak, diberi harga air sekotak RM1.20:

```
while (jum_duit<120) {
    switch(duitmasuk) {
        case satusen:
            jum_duit++;
            break;
        case limasen:
            jum_duit=jum_duit+
            limasen;
            break;
        Case sepuluhsen:
            jum_duit=jum_duit+ sepuluhsen;
            break;
        case samah:
            jum_duit=jum_duit+ samah;
            break;
        case serial:
            jum_duit=jum_duit+ serial;
            break;
    }
}
```