

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



**NAMA : BENIE AGUSTI KAHARAP**  
**NIM : 193030503076**  
**KELAS : (A)**  
**MODUL : I ( DASAR PBO )**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS PALANGKA RAYA**

**2020**

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN II**



Nama : Benie Agusti Kaharap  
NIM : 193030503076  
Kelas : (A)  
Modul : I Dasar PBO

Komposisi	MAX	Nilai
BAB I Tujuan dan Landasan Teori	10	C
BAB II Pembahasan	60	40
BAB III Kesimpulan	20	10
Daftar Pustaka	5	3
Lampiran	5	3
Jumlah	100	62

Penilai

Asisten Praktikum

Diana

## BAB I TUJUAN DAN LANDASAN TEORI

### 1.1. TUJUAN

Setelah menyelesaikan modul ini, mahasiswa diharapkan mampu :

1. Memahami dasar-dasar pemrograman berorientasi obyek.
2. Memahami enkapsulasi.
3. Membuat kelas dan objek.

### 1.2. LANDASAN TEORI

Pemrograman berorientasi objek (object-oriented programming disingkat OOP) merupakan pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Class adalah kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh 'class of dog' adalah suatu unit yang terdiri atas definisi-definisi data - dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi object. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen (sebagaimana kode tersebut digunakan jika tidak menggunakan OOP). Dengan modularitas, struktur dari sebuah program akan terkait dengan aspek-aspek dalam masalah yang akan diselesaikan melalui program tersebut. Cara seperti ini akan menyederhanakan pemetaan dari masalah ke sebuah program ataupun sebaliknya.

Enkapsulasi digunakan untuk memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

Attribute adalah segala sesuatu (data) yang berhubungan dengan karakteristik objek dalam setiap kelas. Tiap atribut memiliki nilai berbeda untuk setiap objek. Method merupakan tingkah laku dari suatu objek, jika bersifat static berarti tingkah laku semua objek dalam kelas tersebut.

Identifier adalah suatu pengenalan atau pengidentifikasi yang kita deklarasikan agar kompiler dapat mengenalinya. Identifier sendiri dapat berupa nama variabel,

*Patang*

fungsi

konstanta fungsi, kelas, template maupun namespace. Biasanya identifier ini terdiri dari 3 bentuk, yaitu privat public dan protected

1. public: anggota pribadi dari kelas dasar hanya dapat diakses oleh anggota yang kelas dasar.
2. private: anggota publik dari kelas dasar dapat diakses oleh anggota yang kelas dasar, anggota kelas turunan serta anggota yang berada di luar kelas dasar dan kelas turunan.
3. protected: anggota dilindungi dari kelas dasar dapat diakses oleh anggota kelas dasar serta anggota kelas turunan.

Constructor adalah anggota class yang bertugas melakukan inisialisasi obyek (instance) dari suatu class C. Constructor mempunyai nama yang sama dengan nama class, dan tidak mempunyai return value. Sebuah class dapat mempunyai lebih dari satu constructor. Constructor yang tidak mempunyai argumen, disebut default constructor, sebaliknya constructor yang mempunyai lebih dari satu argumen adalah non-default constructor. Constructor dengan satu default argument tetap merupakan sebuah default constructor.

diipisah  
wal

## BAB II PEMBAHASAN

### 2.1 Program pertama

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>

int main(void ) {

    // Deklarasi Variable
    char namaBurung[6] = "jalak";
    char jawaban[6] = "-----", huruf[2];
    int kesalahan = 0;
    int i=0, j=0, k=0, benar;

    printf("Pentunjuk Kata Kunci: Nama Burung \n");

    while(kesalahan < 5) {
        printf("Input Karakter: "); scanf ("%s",&huruf);
        benar = 0;
        for(i=0; i<6; i++) {
            if(huruf[i] == namaBurung[i]) {
```

C++

```

        if(jawaban[i] == '-') jawaban[i] = huruf[0]; benar = 1;
    }
}
if(benar!= 1) kesalahan++;
if(strcmp(jawaban,namaBurung)==0) kesalahan = 6;
printf("Kesempatan Menebak: %d\n",5-kesalahan);
printf("Huruf yang sudah ter-tebak: %s\n",jawaban);
}

system("cls");
if(kesalahan == 5) printf("Tebakkan Salah ! \n");
else printf("Tebakkan Benar !\n");

system("pause");
return 0;
}

```

Hasil Output :

```

10 char jawaban[6]
11 int kesalahan =
12 int i=0, j=0, k
13
14 printf("Pentun)
15
16 while(kesalahan
17 printf("Inp
18 benar = 0;
19 for(i=0;i<6
20     if(huru
21     if(
22
23
24 if(benar!=
25 if(strcmp(j
26 printf("Kes
27 printf("Huruf yang sudah ter-tebak: %s\n",jawaban);
28
29

```

2.2 Program kedua :

```

#include<iostream>

using namespace std;

class anjing
{
    public:

```

```
        anjing(int);
        int jlhTelinga();
        int jlhkaki();

    private:
        int telinga;
        int kaki;
};

anjing::anjing(int t)
{
    telinga=t;
    kaki=t;
}

int anjing::jlhTelinga()
{
    return telinga;
}

int anjing::jlhkaki()
{
    return kaki;
}

int main()
{
```

```

anjing t(2);

anjing k(4);

cout<<"=====Modul I===== "<<endl;

cout<<"jumlah telinga anjing adalah : " <<t.jlhTelinga()<<endl;

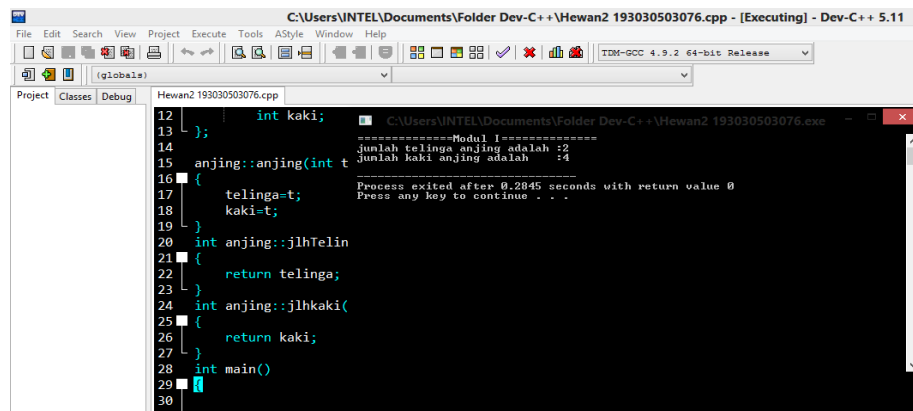
cout<<"jumlah kaki anjing adalah  : " <<k.jlhkaki()<<endl;


return 0;

}

```

Hasil Output :



The screenshot shows the Dev-C++ IDE with the following code in the editor:

```

12 | int kaki;
13 | };
14 |
15 | anjing::anjing(int t
16 | {
17 |     telinga=t;
18 |     kaki=t;
19 | }
20 | int anjing::jlhTelin
21 | {
22 |     return telinga;
23 | }
24 | int anjing::jlhkaki(
25 | {
26 |     return kaki;
27 | }
28 | int main()
29 | {
30 |
31 |     anjing t(2);

```

The output window on the right displays the following text:

```

=====Modul I=====
jumlah telinga anjing adalah :2
jumlah kaki anjing adalah :4
Process exited after 0.2845 seconds with return value 0
Press any key to continue . . .

```

PENJELASAN :

**Pada program 1** saya menggunakan Tebakkan namaBurung, yang saya gunakan adalah integer, dan string yang merupakan fungsi bernilai void, Lalu memasukan fungsi “main” yang merupakan fungsi utama diluar class. Lalu setelah itu memasukan namaBurung, serta angka tebakkan yang telah ditentukan. Dan yang terakhir fungsi printf data agar menampilkan semua data yang telah dimasukan.

**Pada program 2** saya menggunakan nama hewan Anjing, juga menggunakan class dan tipe data yang digunakan string, float, dan void yang juga bersifat public. Yang artinya semuanya dapat diakses diluar class. Lalu memasukan nama, angka dan jumlah angka yang ditampilkan pada Run+compile Setelah itu memasukan fungsi "main" yang merupakan fungsi utama diluar class. Sebenarnya program kedua ini tidak jauh beda dengan program yang pertama.

### BAB III KESIMPULAN

- ☐ Pemrograman berorientasi objek (object-oriented programming disingkat OOP) merupakan pemograman yang berorientasikan kepada objek.
- ☐ Semua data dan fungsi di dalam paradigma dibungkus dalam kelas-kelas atau objek-objek. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya.
- ☐ Class atau kelas merupakan kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu.
- ☐ Fungsi objek adalah untuk membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer, objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
- ☐ Kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti disebut abstraksi.

Paragraf

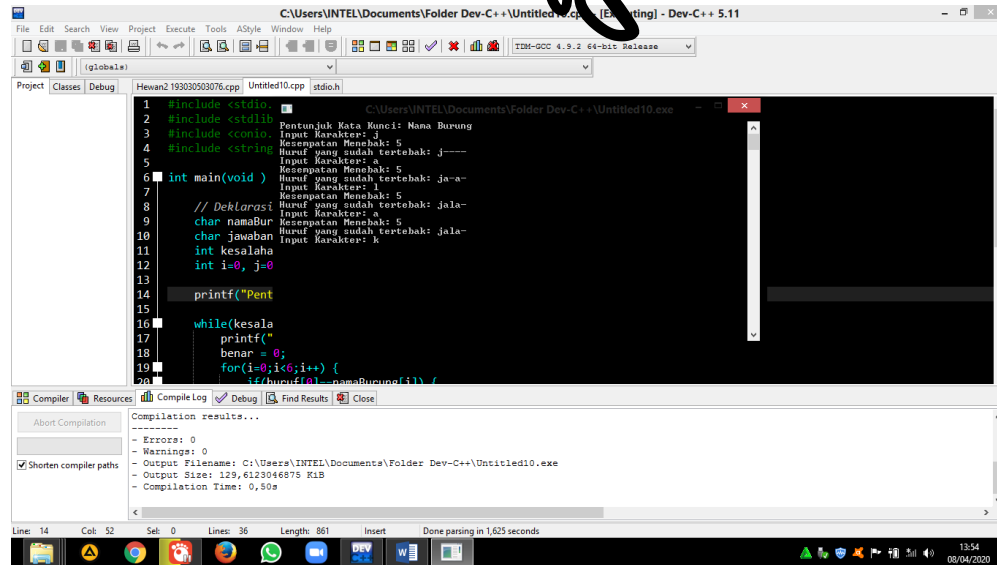
struktur  
pustaka?



Lampiran :

Gambar 1.1 Program 1

nomor  
di bawah  
gambar



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <conio.h>
4 #include <string.h>
5
6 int main(void)
7 {
8     // Deklarasi
9     char namaBur;
10    char jawaban;
11    int kesalahan;
12    int i=0, j=0;
13
14    printf("Pent
15
16    while(kesalahan)
17    {
18        printf("
19        benar = 0;
20        for(i=0; i<strlen(jawaban); i++)
21        {
22            if(jawaban[i] != namaBur[i])
23            {
24                kesalahan++;
25            }
26        }
27    }
28
29    printf("Pent
30
31    while(jawaban[j] != '\0')
32    {
33        printf("
34        benar = 0;
35        for(j=0; j<strlen(jawaban); j++)
36        {
37            if(jawaban[j] != namaBur[j])
38            {
39                kesalahan++;
40            }
41        }
42    }
43
44    printf("Pent
45
46    while(jawaban[j] != '\0')
47    {
48        printf("
49        benar = 0;
50        for(j=0; j<strlen(jawaban); j++)
51        {
52            if(jawaban[j] != namaBur[j])
53            {
54                kesalahan++;
55            }
56        }
57    }
58
59    printf("Pent
60
61    while(jawaban[j] != '\0')
62    {
63        printf("
64        benar = 0;
65        for(j=0; j<strlen(jawaban); j++)
66        {
67            if(jawaban[j] != namaBur[j])
68            {
69                kesalahan++;
70            }
71        }
72    }
73
74    printf("Pent
75
76    while(jawaban[j] != '\0')
77    {
78        printf("
79        benar = 0;
80        for(j=0; j<strlen(jawaban); j++)
81        {
82            if(jawaban[j] != namaBur[j])
83            {
84                kesalahan++;
85            }
86        }
87    }
88
89    printf("Pent
90
91    while(jawaban[j] != '\0')
92    {
93        printf("
94        benar = 0;
95        for(j=0; j<strlen(jawaban); j++)
96        {
97            if(jawaban[j] != namaBur[j])
98            {
99                kesalahan++;
100            }
101        }
102    }
103
104    printf("Pent
105
106    while(jawaban[j] != '\0')
107    {
108        printf("
109        benar = 0;
110        for(j=0; j<strlen(jawaban); j++)
111        {
112            if(jawaban[j] != namaBur[j])
113            {
114                kesalahan++;
115            }
116        }
117    }
118
119    printf("Pent
120
121    while(jawaban[j] != '\0')
122    {
123        printf("
124        benar = 0;
125        for(j=0; j<strlen(jawaban); j++)
126        {
127            if(jawaban[j] != namaBur[j])
128            {
129                kesalahan++;
130            }
131        }
132    }
133
134    printf("Pent
135
136    while(jawaban[j] != '\0')
137    {
138        printf("
139        benar = 0;
140        for(j=0; j<strlen(jawaban); j++)
141        {
142            if(jawaban[j] != namaBur[j])
143            {
144                kesalahan++;
145            }
146        }
147    }
148
149    printf("Pent
150
151    while(jawaban[j] != '\0')
152    {
153        printf("
154        benar = 0;
155        for(j=0; j<strlen(jawaban); j++)
156        {
157            if(jawaban[j] != namaBur[j])
158            {
159                kesalahan++;
160            }
161        }
162    }
163
164    printf("Pent
165
166    while(jawaban[j] != '\0')
167    {
168        printf("
169        benar = 0;
170        for(j=0; j<strlen(jawaban); j++)
171        {
172            if(jawaban[j] != namaBur[j])
173            {
174                kesalahan++;
175            }
176        }
177    }
178
179    printf("Pent
180
181    while(jawaban[j] != '\0')
182    {
183        printf("
184        benar = 0;
185        for(j=0; j<strlen(jawaban); j++)
186        {
187            if(jawaban[j] != namaBur[j])
188            {
189                kesalahan++;
190            }
191        }
192    }
193
194    printf("Pent
195
196    while(jawaban[j] != '\0')
197    {
198        printf("
199        benar = 0;
200        for(j=0; j<strlen(jawaban); j++)
201        {
202            if(jawaban[j] != namaBur[j])
203            {
204                kesalahan++;
205            }
206        }
207    }
208
209    printf("Pent
210
211    while(jawaban[j] != '\0')
212    {
213        printf("
214        benar = 0;
215        for(j=0; j<strlen(jawaban); j++)
216        {
217            if(jawaban[j] != namaBur[j])
218            {
219                kesalahan++;
220            }
221        }
222    }
223
224    printf("Pent
225
226    while(jawaban[j] != '\0')
227    {
228        printf("
229        benar = 0;
230        for(j=0; j<strlen(jawaban); j++)
231        {
232            if(jawaban[j] != namaBur[j])
233            {
234                kesalahan++;
235            }
236        }
237    }
238
239    printf("Pent
240
241    while(jawaban[j] != '\0')
242    {
243        printf("
244        benar = 0;
245        for(j=0; j<strlen(jawaban); j++)
246        {
247            if(jawaban[j] != namaBur[j])
248            {
249                kesalahan++;
250            }
251        }
252    }
253
254    printf("Pent
255
256    while(jawaban[j] != '\0')
257    {
258        printf("
259        benar = 0;
260        for(j=0; j<strlen(jawaban); j++)
261        {
262            if(jawaban[j] != namaBur[j])
263            {
264                kesalahan++;
265            }
266        }
267    }
268
269    printf("Pent
270
271    while(jawaban[j] != '\0')
272    {
273        printf("
274        benar = 0;
275        for(j=0; j<strlen(jawaban); j++)
276        {
277            if(jawaban[j] != namaBur[j])
278            {
279                kesalahan++;
280            }
281        }
282    }
283
284    printf("Pent
285
286    while(jawaban[j] != '\0')
287    {
288        printf("
289        benar = 0;
290        for(j=0; j<strlen(jawaban); j++)
291        {
292            if(jawaban[j] != namaBur[j])
293            {
294                kesalahan++;
295            }
296        }
297    }
298
299    printf("Pent
300
301    while(jawaban[j] != '\0')
302    {
303        printf("
304        benar = 0;
305        for(j=0; j<strlen(jawaban); j++)
306        {
307            if(jawaban[j] != namaBur[j])
308            {
309                kesalahan++;
310            }
311        }
312    }
313
314    printf("Pent
315
316    while(jawaban[j] != '\0')
317    {
318        printf("
319        benar = 0;
320        for(j=0; j<strlen(jawaban); j++)
321        {
322            if(jawaban[j] != namaBur[j])
323            {
324                kesalahan++;
325            }
326        }
327    }
328
329    printf("Pent
330
331    while(jawaban[j] != '\0')
332    {
333        printf("
334        benar = 0;
335        for(j=0; j<strlen(jawaban); j++)
336        {
337            if(jawaban[j] != namaBur[j])
338            {
339                kesalahan++;
340            }
341        }
342    }
343
344    printf("Pent
345
346    while(jawaban[j] != '\0')
347    {
348        printf("
349        benar = 0;
350        for(j=0; j<strlen(jawaban); j++)
351        {
352            if(jawaban[j] != namaBur[j])
353            {
354                kesalahan++;
355            }
356        }
357    }
358
359    printf("Pent
360
361    while(jawaban[j] != '\0')
362    {
363        printf("
364        benar = 0;
365        for(j=0; j<strlen(jawaban); j++)
366        {
367            if(jawaban[j] != namaBur[j])
368            {
369                kesalahan++;
370            }
371        }
372    }
373
374    printf("Pent
375
376    while(jawaban[j] != '\0')
377    {
378        printf("
379        benar = 0;
380        for(j=0; j<strlen(jawaban); j++)
381        {
382            if(jawaban[j] != namaBur[j])
383            {
384                kesalahan++;
385            }
386        }
387    }
388
389    printf("Pent
390
391    while(jawaban[j] != '\0')
392    {
393        printf("
394        benar = 0;
395        for(j=0; j<strlen(jawaban); j++)
396        {
397            if(jawaban[j] != namaBur[j])
398            {
399                kesalahan++;
400            }
401        }
402    }
403
404    printf("Pent
405
406    while(jawaban[j] != '\0')
407    {
408        printf("
409        benar = 0;
410        for(j=0; j<strlen(jawaban); j++)
411        {
412            if(jawaban[j] != namaBur[j])
413            {
414                kesalahan++;
415            }
416        }
417    }
418
419    printf("Pent
420
421    while(jawaban[j] != '\0')
422    {
423        printf("
424        benar = 0;
425        for(j=0; j<strlen(jawaban); j++)
426        {
427            if(jawaban[j] != namaBur[j])
428            {
429                kesalahan++;
430            }
431        }
432    }
433
434    printf("Pent
435
436    while(jawaban[j] != '\0')
437    {
438        printf("
439        benar = 0;
440        for(j=0; j<strlen(jawaban); j++)
441        {
442            if(jawaban[j] != namaBur[j])
443            {
444                kesalahan++;
445            }
446        }
447    }
448
449    printf("Pent
450
451    while(jawaban[j] != '\0')
452    {
453        printf("
454        benar = 0;
455        for(j=0; j<strlen(jawaban); j++)
456        {
457            if(jawaban[j] != namaBur[j])
458            {
459                kesalahan++;
460            }
461        }
462    }
463
464    printf("Pent
465
466    while(jawaban[j] != '\0')
467    {
468        printf("
469        benar = 0;
470        for(j=0; j<strlen(jawaban); j++)
471        {
472            if(jawaban[j] != namaBur[j])
473            {
474                kesalahan++;
475            }
476        }
477    }
478
479    printf("Pent
480
481    while(jawaban[j] != '\0')
482    {
483        printf("
484        benar = 0;
485        for(j=0; j<strlen(jawaban); j++)
486        {
487            if(jawaban[j] != namaBur[j])
488            {
489                kesalahan++;
490            }
491        }
492    }
493
494    printf("Pent
495
496    while(jawaban[j] != '\0')
497    {
498        printf("
499        benar = 0;
500        for(j=0; j<strlen(jawaban); j++)
501        {
502            if(jawaban[j] != namaBur[j])
503            {
504                kesalahan++;
505            }
506        }
507    }
508
509    printf("Pent
510
511    while(jawaban[j] != '\0')
512    {
513        printf("
514        benar = 0;
515        for(j=0; j<strlen(jawaban); j++)
516        {
517            if(jawaban[j] != namaBur[j])
518            {
519                kesalahan++;
520            }
521        }
522    }
523
524    printf("Pent
525
526    while(jawaban[j] != '\0')
527    {
528        printf("
529        benar = 0;
530        for(j=0; j<strlen(jawaban); j++)
531        {
532            if(jawaban[j] != namaBur[j])
533            {
534                kesalahan++;
535            }
536        }
537    }
538
539    printf("Pent
540
541    while(jawaban[j] != '\0')
542    {
543        printf("
544        benar = 0;
545        for(j=0; j<strlen(jawaban); j++)
546        {
547            if(jawaban[j] != namaBur[j])
548            {
549                kesalahan++;
550            }
551        }
552    }
553
554    printf("Pent
555
556    while(jawaban[j] != '\0')
557    {
558        printf("
559        benar = 0;
560        for(j=0; j<strlen(jawaban); j++)
561        {
562            if(jawaban[j] != namaBur[j])
563            {
564                kesalahan++;
565            }
566        }
567    }
568
569    printf("Pent
570
571    while(jawaban[j] != '\0')
572    {
573        printf("
574        benar = 0;
575        for(j=0; j<strlen(jawaban); j++)
576        {
577            if(jawaban[j] != namaBur[j])
578            {
579                kesalahan++;
580            }
581        }
582    }
583
584    printf("Pent
585
586    while(jawaban[j] != '\0')
587    {
588        printf("
589        benar = 0;
590        for(j=0; j<strlen(jawaban); j++)
591        {
592            if(jawaban[j] != namaBur[j])
593            {
594                kesalahan++;
595            }
596        }
597    }
598
599    printf("Pent
600
601    while(jawaban[j] != '\0')
602    {
603        printf("
604        benar = 0;
605        for(j=0; j<strlen(jawaban); j++)
606        {
607            if(jawaban[j] != namaBur[j])
608            {
609                kesalahan++;
610            }
611        }
612    }
613
614    printf("Pent
615
616    while(jawaban[j] != '\0')
617    {
618        printf("
619        benar = 0;
620        for(j=0; j<strlen(jawaban); j++)
621        {
622            if(jawaban[j] != namaBur[j])
623            {
624                kesalahan++;
625            }
626        }
627    }
628
629    printf("Pent
630
631    while(jawaban[j] != '\0')
632    {
633        printf("
634        benar = 0;
635        for(j=0; j<strlen(jawaban); j++)
636        {
637            if(jawaban[j] != namaBur[j])
638            {
639                kesalahan++;
640            }
641        }
642    }
643
644    printf("Pent
645
646    while(jawaban[j] != '\0')
647    {
648        printf("
649        benar = 0;
650        for(j=0; j<strlen(jawaban); j++)
651        {
652            if(jawaban[j] != namaBur[j])
653            {
654                kesalahan++;
655            }
656        }
657    }
658
659    printf("Pent
660
661    while(jawaban[j] != '\0')
662    {
663        printf("
664        benar = 0;
665        for(j=0; j<strlen(jawaban); j++)
666        {
667            if(jawaban[j] != namaBur[j])
668            {
669                kesalahan++;
670            }
671        }
672    }
673
674    printf("Pent
675
676    while(jawaban[j] != '\0')
677    {
678        printf("
679        benar = 0;
680        for(j=0; j<strlen(jawaban); j++)
681        {
682            if(jawaban[j] != namaBur[j])
683            {
684                kesalahan++;
685            }
686        }
687    }
688
689    printf("Pent
690
691    while(jawaban[j] != '\0')
692    {
693        printf("
694        benar = 0;
695        for(j=0; j<strlen(jawaban); j++)
696        {
697            if(jawaban[j] != namaBur[j])
698            {
699                kesalahan++;
700            }
701        }
702    }
703
704    printf("Pent
705
706    while(jawaban[j] != '\0')
707    {
708        printf("
709        benar = 0;
710        for(j=0; j<strlen(jawaban); j++)
711        {
712            if(jawaban[j] != namaBur[j])
713            {
714                kesalahan++;
715            }
716        }
717    }
718
719    printf("Pent
720
721    while(jawaban[j] != '\0')
722    {
723        printf("
724        benar = 0;
725        for(j=0; j<strlen(jawaban); j++)
726        {
727            if(jawaban[j] != namaBur[j])
728            {
729                kesalahan++;
730            }
731        }
732    }
733
734    printf("Pent
735
736    while(jawaban[j] != '\0')
737    {
738        printf("
739        benar = 0;
740        for(j=0; j<strlen(jawaban); j++)
741        {
742            if(jawaban[j] != namaBur[j])
743            {
744                kesalahan++;
745            }
746        }
747    }
748
749    printf("Pent
750
751    while(jawaban[j] != '\0')
752    {
753        printf("
754        benar = 0;
755        for(j=0; j<strlen(jawaban); j++)
756        {
757            if(jawaban[j] != namaBur[j])
758            {
759                kesalahan++;
760            }
761        }
762    }
763
764    printf("Pent
765
766    while(jawaban[j] != '\0')
767    {
768        printf("
769        benar = 0;
770        for(j=0; j<strlen(jawaban); j++)
771        {
772            if(jawaban[j] != namaBur[j])
773            {
774                kesalahan++;
775            }
776        }
777    }
778
779    printf("Pent
780
781    while(jawaban[j] != '\0')
782    {
783        printf("
784        benar = 0;
785        for(j=0; j<strlen(jawaban); j++)
786        {
787            if(jawaban[j] != namaBur[j])
788            {
789                kesalahan++;
790            }
791        }
792    }
793
794    printf("Pent
795
796    while(jawaban[j] != '\0')
797    {
798        printf("
799        benar = 0;
800        for(j=0; j<strlen(jawaban); j++)
801        {
802            if(jawaban[j] != namaBur[j])
803            {
804                kesalahan++;
805            }
806        }
807    }
808
809    printf("Pent
810
811    while(jawaban[j] != '\0')
812    {
813        printf("
814        benar = 0;
815        for(j=0; j<strlen(jawaban); j++)
816        {
817            if(jawaban[j] != namaBur[j])
818            {
819                kesalahan++;
820            }
821        }
822    }
823
824    printf("Pent
825
826    while(jawaban[j] != '\0')
827    {
828        printf("
829        benar = 0;
830        for(j=0; j<strlen(jawaban); j++)
831        {
832            if(jawaban[j] != namaBur[j])
833            {
834                kesalahan++;
835            }
836        }
837    }
838
839    printf("Pent
840
841    while(jawaban[j] != '\0')
842    {
843        printf("
844        benar = 0;
845        for(j=0; j<strlen(jawaban); j++)
846        {
847            if(jawaban[j] != namaBur[j])
848            {
849                kesalahan++;
850            }
851        }
852    }
853
854    printf("Pent
855
856    while(jawaban[j] != '\0')
857    {
858        printf("
859        benar = 0;
860        for(j=0; j<strlen(jawaban); j++)
861        {
862            if(jawaban[j] != namaBur[j])
863            {
864                kesalahan++;
865            }
866        }
867    }
868
869    printf("Pent
870
871    while(jawaban[j] != '\0')
872    {
873        printf("
874        benar = 0;
875        for(j=0; j<strlen(jawaban); j++)
876        {
877            if(jawaban[j] != namaBur[j])
878            {
879                kesalahan++;
880            }
881        }
882    }
883
884    printf("Pent
885
886    while(jawaban[j] != '\0')
887    {
888        printf("
889        benar = 0;
890        for(j=0; j<strlen(jawaban); j++)
891        {
892            if(jawaban[j] != namaBur[j])
893            {
894                kesalahan++;
895            }
896        }
897    }
898
899    printf("Pent
900
901    while(jawaban[j] != '\0')
902    {
903        printf("
904        benar = 0;
905        for(j=0; j<strlen(jawaban); j++)
906        {
907            if(jawaban[j] != namaBur[j])
908            {
909                kesalahan++;
910            }
911        }
912    }
913
914    printf("Pent
915
916    while(jawaban[j] != '\0')
917    {
918        printf("
919        benar = 0;
920        for(j=0; j<strlen(jawaban); j++)
921        {
922            if(jawaban[j] != namaBur[j])
923            {
924                kesalahan++;
925            }
926        }
927    }
928
929    printf("Pent
930
931    while(jawaban[j] != '\0')
932    {
933        printf("
934        benar = 0;
935        for(j=0; j<strlen(jawaban); j++)
936        {
937            if(jawaban[j] != namaBur[j])
938            {
939                kesalahan++;
940            }
941        }
942    }
943
944    printf("Pent
945
946    while(jawaban[j] != '\0')
947    {
948        printf("
949        benar = 0;
950        for(j=0; j<strlen(jawaban); j++)
951        {
952            if(jawaban[j] != namaBur[j])
953            {
954                kesalahan++;
955            }
956        }
957    }
958
959    printf("Pent
960
961    while(jawaban[j] != '\0')
962    {
963        printf("
964        benar = 0;
965        for(j=0; j<strlen(jawaban); j++)
966        {
967            if(jawaban[j] != namaBur[j])
968            {
969                kesalahan++;
970            }
971        }
972    }
973
974    printf("Pent
975
976    while(jawaban[j] != '\0')
977    {
978        printf("
979        benar = 0;
980        for(j=0; j<strlen(jawaban); j++)
981        {
982            if(jawaban[j] != namaBur[j])
983            {
984                kesalahan++;
985            }
986        }
987    }
988
989    printf("Pent
990
991    while(jawaban[j] != '\0')
992    {
993        printf("
994        benar = 0;
995        for(j=0; j<strlen(jawaban); j++)
996        {
997            if(jawaban[j] != namaBur[j])
998            {
999                kesalahan++;
1000            }
1001        }
1002    }
1003
1004    printf("Pent
1005
1006    while(jawaban[j] != '\0')
1007    {
1008        printf("
1009        benar = 0;
1010        for(j=0; j<strlen(jawaban); j++)
1011        {
1012            if(jawaban[j] != namaBur[j])
1013            {
1014                kesalahan++;
1015            }
1016        }
1017    }
1018
1019    printf("Pent
1020
1021    while(jawaban[j] != '\0')
1022    {
1023        printf("
1024        benar = 0;
1025        for(j=0; j<strlen(jawaban); j++)
1026        {
1027            if(jawaban[j] != namaBur[j])
1028            {
1029                kesalahan++;
1030            }
1031        }
1032    }
1033
1034    printf("Pent
1035
1036    while(jawaban[j] != '\0')
1037    {
1038        printf("
1039        benar = 0;
1040        for(j=0; j<strlen(jawaban); j++)
1041        {
1042            if(jawaban[j] != namaBur[j])
1043            {
1044                kesalahan++;
1045            }
1046        }
1047    }
1048
1049    printf("Pent
1050
1051    while(jawaban[j] != '\0')
1052    {
1053        printf("
1054        benar = 0;
1055        for(j=0; j<strlen(jawaban); j++)
1056        {
1057            if(jawaban[j] != namaBur[j])
1058            {
1059                kesalahan++;
1060            }
1061        }
1062    }
1063
1064    printf("Pent
1065
1066    while(jawaban[j] != '\0')
1067    {
1068        printf("
1069        benar = 0;
1070        for(j=0; j<strlen(jawaban); j++)
1071        {
1072            if(jawaban[j] != namaBur[j])
1073            {
1074                kesalahan++;
1075            }
1076        }
1077    }
1078
1079    printf("Pent
1080
1081    while(jawaban[j] != '\0')
1082    {
1083        printf("
1084        benar = 0;
1085        for(j=0; j<strlen(jawaban); j++)
1086        {
1087            if(jawaban[j] != namaBur[j])
1088            {
1089                kesalahan++;
1090            }
1091        }
1092    }
1093
1094    printf("Pent
1095
1096    while(jawaban[j] != '\0')
1097    {
1098        printf("
1099        benar = 0;
1100        for(j=0; j<strlen(jawaban); j++)
1101        {
1102            if(jawaban[j] != namaBur[j])
1103            {
1104                kesalahan++;
1105            }
1106        }
1107    }
1108
1109    printf("Pent
1110
1111    while(jawaban[j] != '\0')
1112    {
1113        printf("
1114        benar = 0;
1115        for(j=0; j<strlen(jawaban); j++)
1116        {
1117            if(jawaban[j] != namaBur[j])
1118            {
1119                kesalahan++;
1120            }
1121        }
1122    }
1123
1124    printf("Pent
1125
1126    while(jawaban[j] != '\0')
1127    {
1128        printf("
1129        benar = 0;
1130        for(j=0; j<strlen(jawaban); j++)
1131        {
1132            if(jawaban[j] != namaBur[j])
1133            {
1134                kesalahan++;
1135            }
1136        }
1137    }
1138
1139    printf("Pent
1140
1141    while(jawaban[j] != '\0')
1142    {
1143        printf("
1144        benar = 0;
1145        for(j=0; j<strlen(jawaban); j++)
1146        {
1147            if(jawaban[j] != namaBur[j])
1148            {
1149                kesalahan++;
1150            }
1151        }
1152    }
1153
1154    printf("Pent
1155
1156    while(jawaban[j] != '\0')
1157    {
1158        printf("
1159        benar = 0;
1160        for(j=0; j<strlen(jawaban); j++)
1161        {
1162            if(jawaban[j] != namaBur[j])
1163            {
1164                kesalahan++;
1165            }
1166        }
1167    }
1168
1169    printf("Pent
1170
1171    while(jawaban[j] != '\0')
1172    {
1173        printf("
1174        benar = 0;
1175        for(j=0; j<strlen(jawaban); j++)
1176        {
1177            if(jawaban[j] != namaBur[j])
1178            {
1179                kesalahan++;
1180            }
1181        }
1182    }
1183
1184    printf("Pent
1185
1186    while(jawaban[j] != '\0')
1187    {
1188        printf("
1189        benar = 0;
1190        for(j=0; j<strlen(jawaban); j++)
1191        {
1192            if(jawaban[j] != namaBur[j])
1193            {
1194                kesalahan++;
1195            }
1196        }
1197    }
1198
1199    printf("Pent
1200
1201    while(jawaban[j] != '\0')
1202    {
1203        printf("
1204        benar = 0;
1205        for(j=0; j<strlen(jawaban); j++)
1206        {
1207            if(jawaban[j] != namaBur[j])
1208            {
1209                kesalahan++;
1210            }
1211        }
1212    }
1213
1214    printf("Pent
1215
1216    while(jawaban[j] != '\0')
1217    {
1218        printf("
1219        benar = 0;
1220        for(j=0; j<strlen(jawaban); j++)
1221        {
1222            if(jawaban[j] != namaBur[j])
1223            {
1224                kesalahan++;
1225            }
1226        }
1227    }
1228
1229    printf("Pent
1230
1231    while(jawaban[j] != '\0')
1232    {
1233        printf("
1234        benar = 0;
1235        for(j=0; j<strlen(jawaban); j++)
1236        {
1237            if(jawaban[j] != namaBur[j])
1238            {
1239                kesalahan++;
1240            }
1241        }
1242    }
1243
1244    printf("Pent
1245
1246    while(jawaban[j] != '\0')
1247    {
1248        printf("
1249        benar = 0;
1250        for(j=0; j<strlen(jawaban); j++)
1251        {
1252            if(jawaban[j] != namaBur[j])
1253            {
1254                kesalahan++;
1255            }
1256        }
1257    }
1258
1259    printf("Pent
1260
1261    while(jawaban[j] != '\0')
1262    {
1263        printf("
1264        benar = 0;
1265        for(j=0; j<strlen(jawaban); j++)
1266        {
1267            if(jawaban[j] != namaBur[j])
1268            {
1269                kesalahan++;
1270            }
1271        }
1272    }
1273
1274    printf("Pent
1275
1276    while(jawaban[j] != '\0')
1277    {
1278        printf("
1279        benar = 0;
1280        for(j=0; j<strlen(jawaban); j++)
1281        {
1282            if(jawaban[j] != namaBur[j])
1283            {
1284                kesalahan++;
1285            }
1286        }
1287    }
1288
1289    printf("Pent
1290
1291    while(jawaban[j] != '\0')
1292    {
1293        printf("
1294        benar = 0;
1295        for(j=0; j<strlen(jawaban); j++)
1296        {
1297            if(jawaban[j] != namaBur[j])
1298            {
1299                kesalahan++;
1300            }
1301        }
1302    }
1303
1304    printf("Pent
1305
1306    while(jawaban[j] != '\0')
1307    {
1308        printf("
1309        benar = 0;
1310        for(j=0; j<strlen(jawaban); j++)
1311        {
1312            if(jawaban[j] != namaBur[j])
1313            {
1314                kesalahan++;
1315            }
1316        }
1317    }
1318
1319    printf("Pent
1320
1321    while(jawaban[j] != '\0')
1322    {
1323        printf("
1324        benar = 0;
1325        for(j=0; j<strlen(jawaban); j++)
1326        {
1327            if(jawaban[j] != namaBur[j])
1328            {
1329                kesalahan++;
1330            }
1331        }
1332    }
1333
1334    printf("Pent
1335
1336    while(jawaban[j] != '\0')
1337    {
1338        printf("
1339        benar = 0;
1340        for(j=0; j<strlen(jawaban); j++)
1341        {
1342            if(jawaban[j] != namaBur[j])
1343            {
1344                kesalahan++;
1345            }
1346        }
1347    }
1348
1349    printf("Pent
1350
1351    while(jawaban[j] != '\0')
1352    {
1353        printf("
1354        benar = 0;
1355        for(j=0; j<strlen(jawaban); j++)
1356        {
1357            if(jawaban[j] != namaBur[j])
1358            {
1359                kesalahan++;
1360            }
1361        }
1362    }
1363
1364    printf("Pent
1365
1366    while(jawaban[j] != '\0')
1367    {
1368        printf("
1369        benar = 0;
1370        for(j=0; j<strlen(jawaban); j++)
1371        {
1372            if(jawaban[j] != namaBur[j])
1373            {
1374                kesalahan++;
1375            }
1376        }
1377    }
1378
1379    printf("Pent
1380
1381    while(jawaban[j] != '\0')
1
```