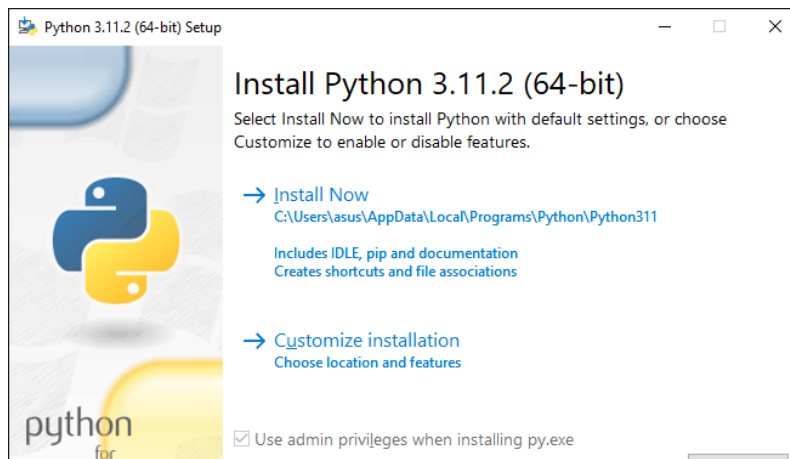
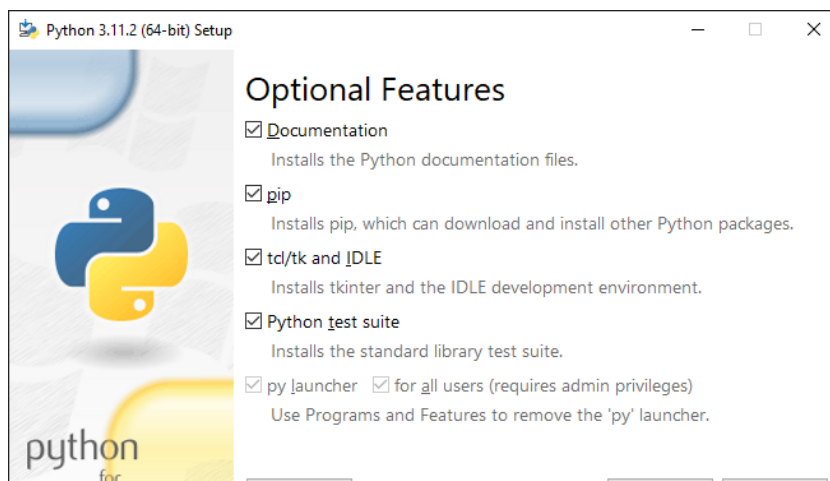


PROSES INSTALASI PYTHON

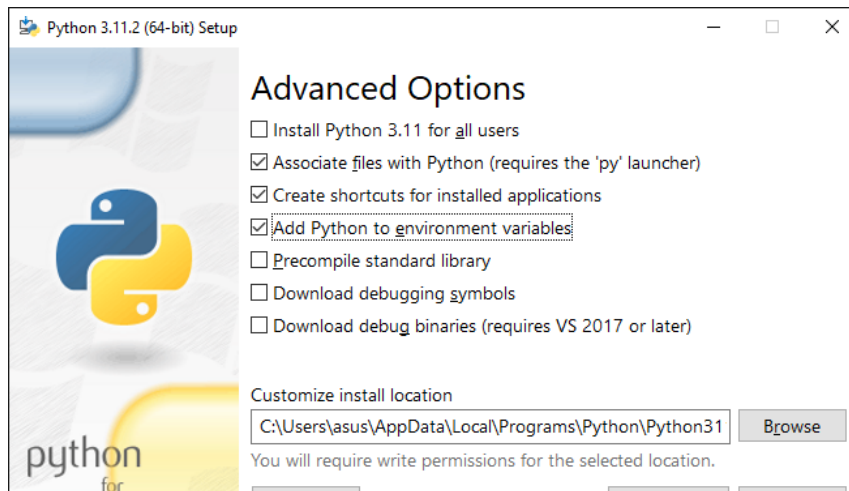
1. Pertama pilih versi Python yang sesuai dengan tipe sistem operasi yang digunakan, kemudian download dan double klik. Setelah itu akan muncul seperti gambar dibawah, selanjutnya pilih Customize installation.



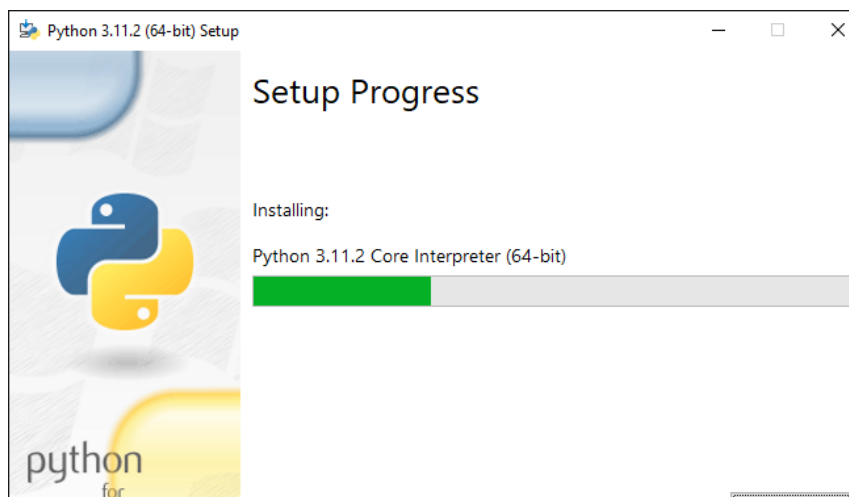
2. Kemudian klik next untuk melanjutkan proses instalasi.



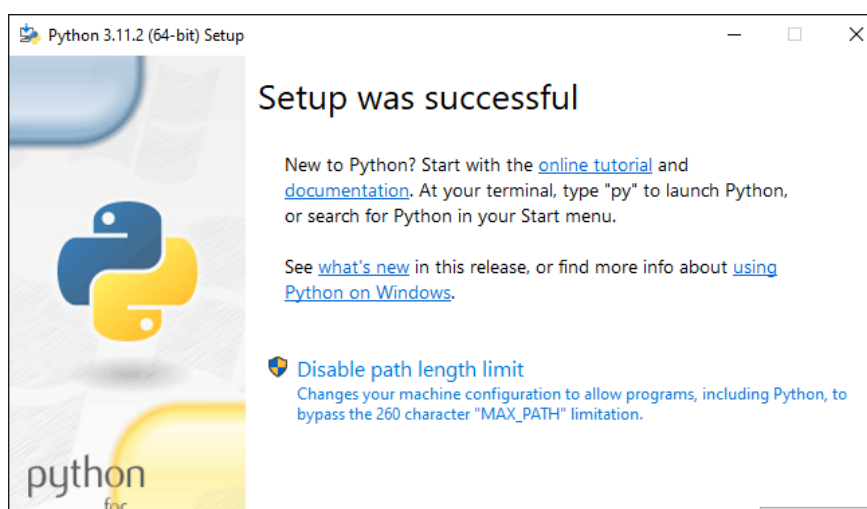
3. Selanjutnya atur direktori atau lokasi dimana Anda ingin menginstal Python, kemudian klik Install.



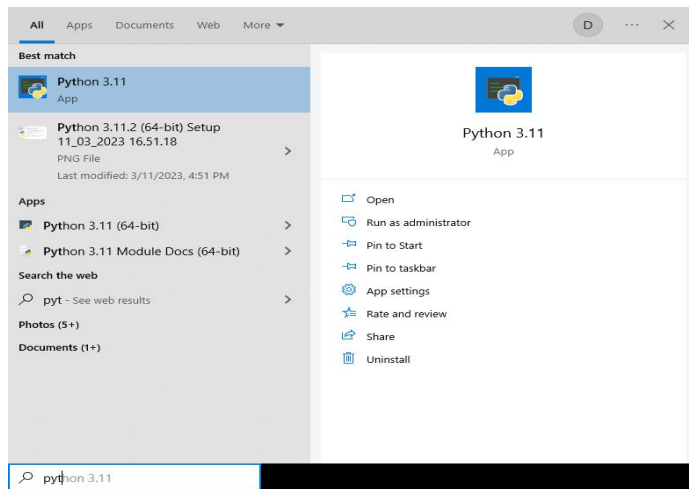
4. Tunggu hingga proses selesai.



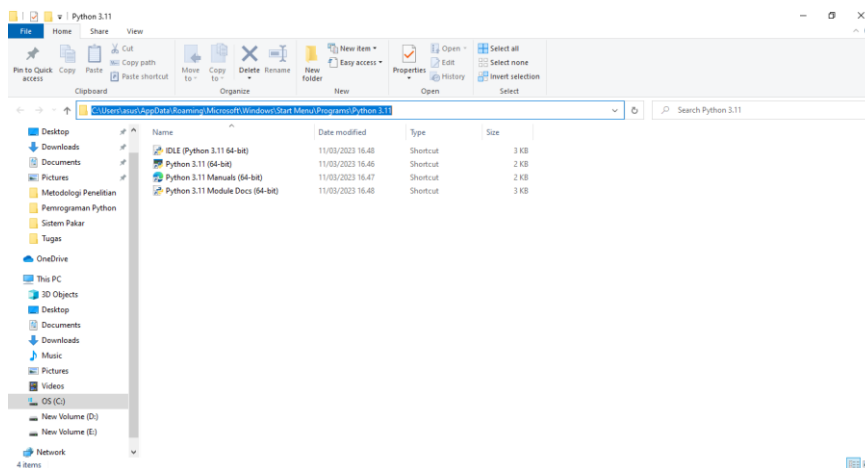
5. Setelah selesai dan sukses, klik Close.



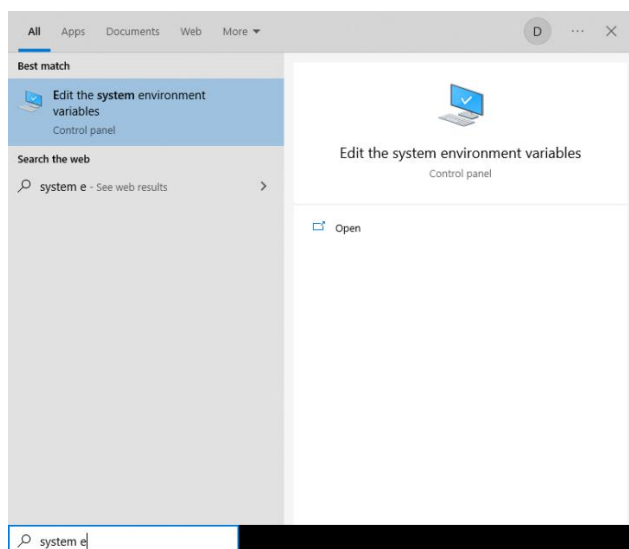
6. Kemudian kita bisa mengecek apakah Python sudah diinstal dengan melakukan searching atau pencarian di Windows.



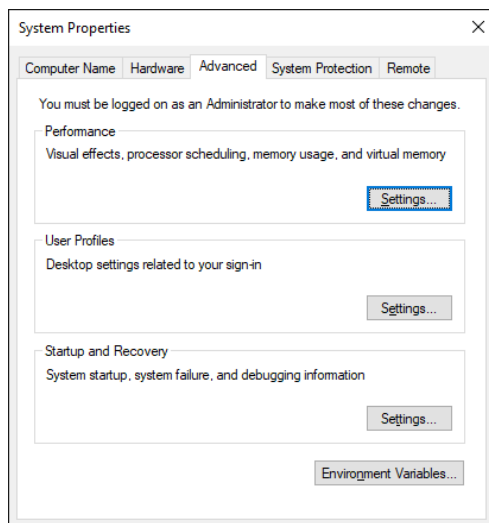
7. Kemudian copy alamat direktori Python sebagai lokasi mensetting Environment



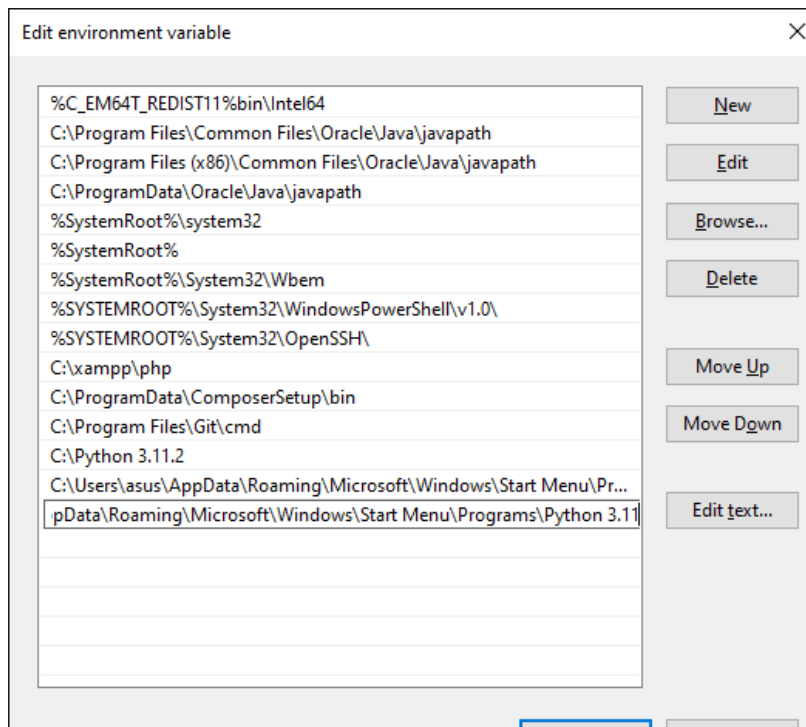
8. Kemudian buka System Environment Variables untuk men-setting Path. Buka Control Panel lalu cari System Environment Variables atau bisa langsung search melalui pencarian Windows seperti gambar dibawah.



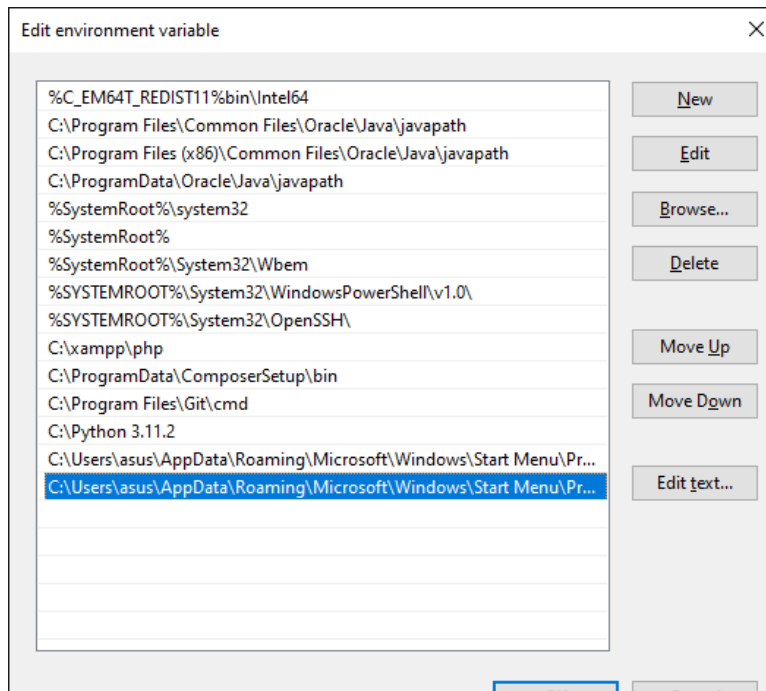
9. Setelah muncul seperti gambar dibawah klik Environment Variables.



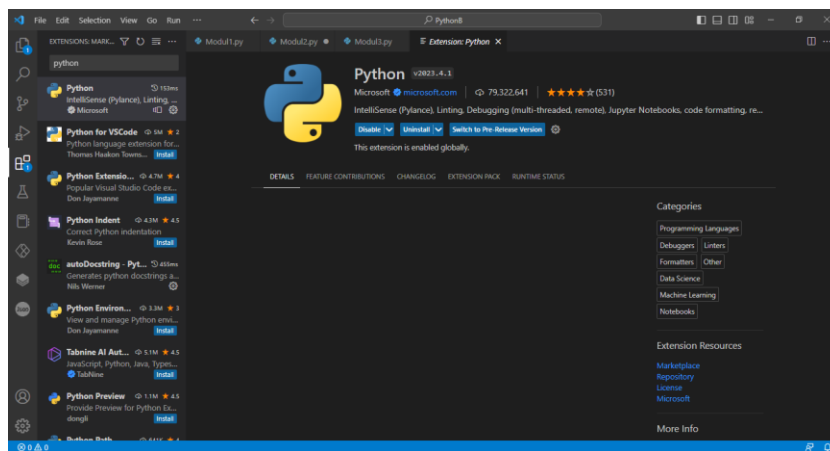
10. Pilih bagian System variables Path lalu klik Edit seperti pada gambar dibawah.



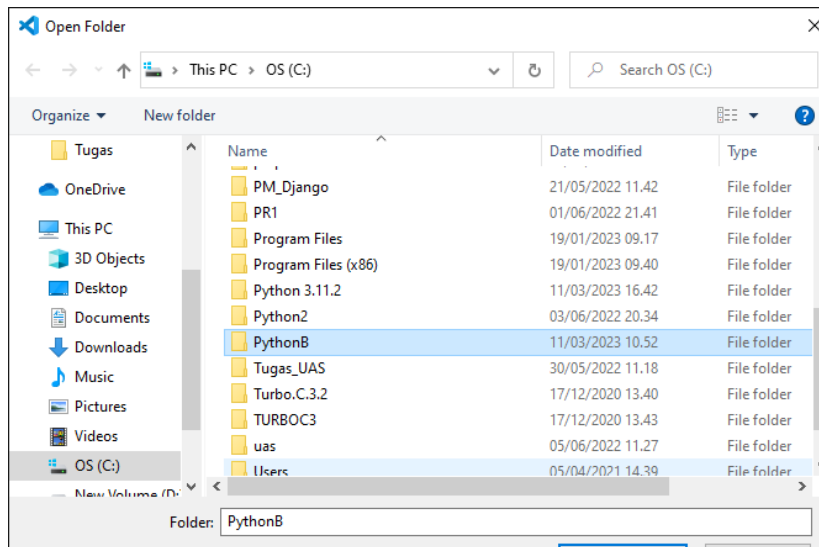
11. Klik tombol New kemudian paste alamat yang telah di copy sebelumnya sebagai lokasi, setelah selesai klik OK.



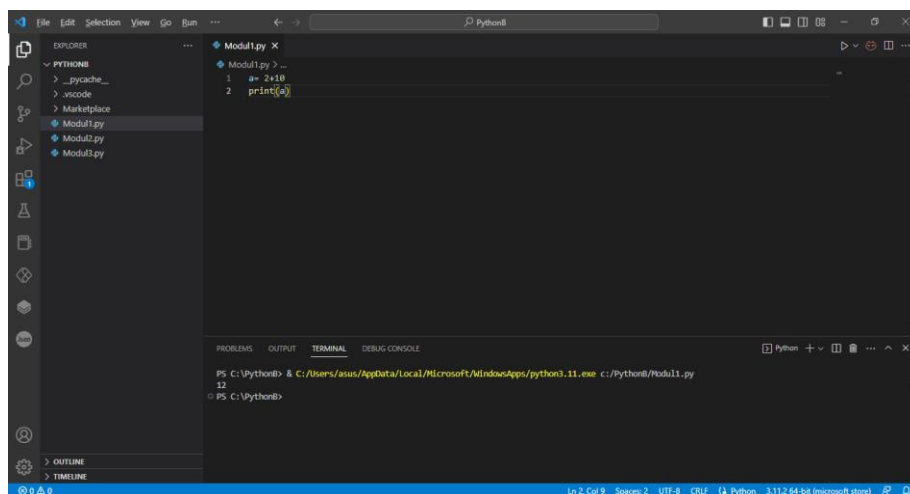
12. Kemudian buka VS Code yang sudah terinstal, ,alu klik menu Extensions pada Activity Bar (sebelah kiri). Ketik python, kemudian Install dan reload VS Code, seperti pada gambar dibawah.



13. Pada bagian Side Bar, klik Open Folder untuk membuka folder untuk menjalankan folder proyek kemudian select folder, seperti contoh dibawah.



14. Setelah folder tampil. Klik New File untuk membuat file Python, Ketik nama file dengan ekstensi.py dibelakang nama file. Dan pilih menu Debug pada Activity Bar, kemudian jalankan project. Sepeti contoh dibawah file dengan nama modul1.py akan menampilkan penjumlahan sederhana seperti pada gambar dibawah.



MODUL 1

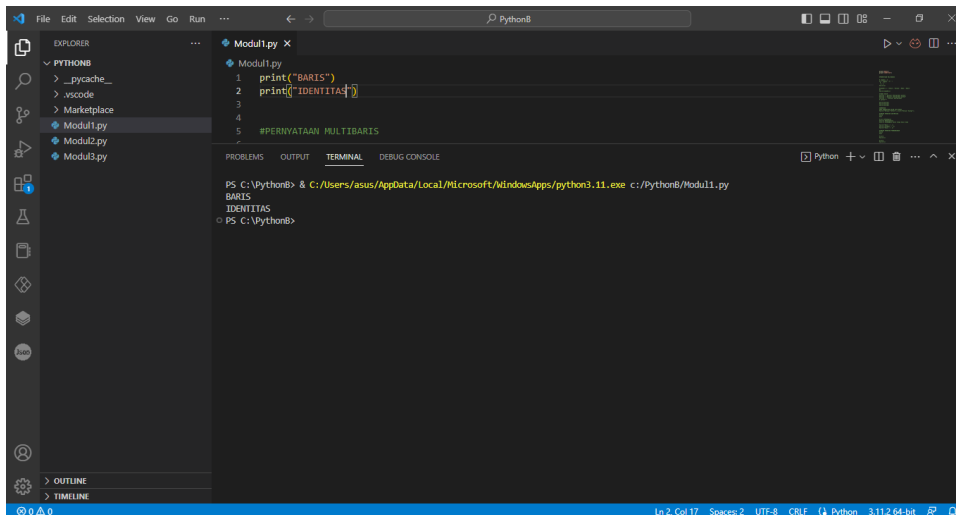
1. Identifier

Merupakan identitas atau nama yang telah diberikan kepada function, variabel, obyek, class, namespace dan lain-lain.

2. Baris dan Indentasi

Python tidak menggunakan tanda { } untuk menandai blok atau grup kode di python menggunakan tanda indentasi (spasi). Jumlah spasi untuk setiap baris yang ada dalam satu blok kode harus sama.

Contoh program baris dan identitas yang benar :

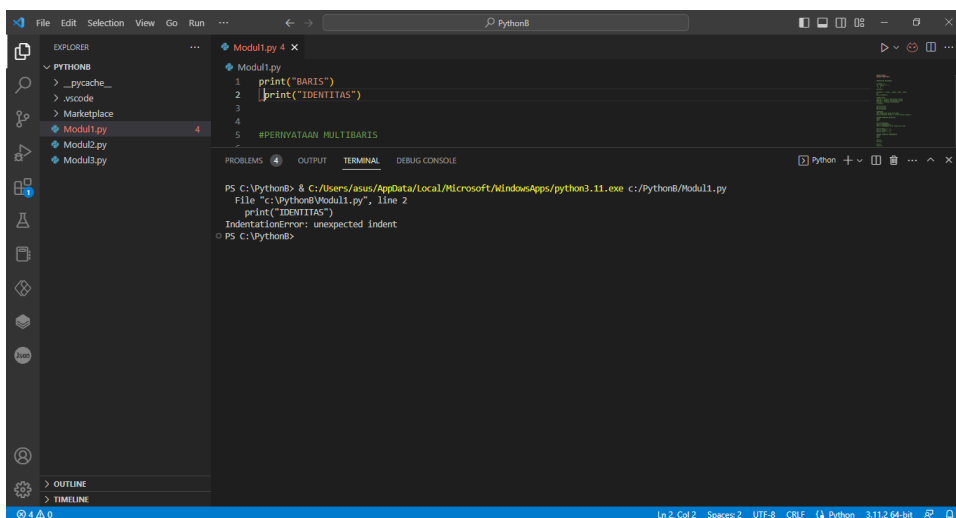


```
Modul1.py X
1 print("BARIS")
2 print("IDENTITAS")
3
4
5 #PERNYATAAN MULTIBARIS
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul1.py
BARIS
IDENTITAS
PS C:\Python>
```

Contoh program salah baris dan identitas :



```
Modul1.py 4 X
1 print("BARIS")
2 print("IDENTITAS")
3
4
5 #PERNYATAAN MULTIBARIS
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul1.py
File "c:\python8\Modul1.py", line 2
    print("IDENTITAS")
    ^
IndentationError: unexpected indent
PS C:\Python>
```

3. Pernyataan Multibaris

Digunakan untuk membuat pernyataan yang terdiri dari beberapa baris menggunakan tanda backslash (\).

Contoh program pernyataan multibaris:

```
5 #PERNYATAAN MULTIBARIS
6 angka1 = 2
7 a = angka1 + 4 + \
8     7 + \
9     5
10 print(a)
```

The terminal shows the output of the script: 18.

Contoh program statement multibaris dalam tanda kurung :

```
11 #print(a)
12
13 namaHari = ['Senin', 'Selasa', 'Rabu', 'Kamis']
14
15 print(namaHari)
```

The terminal shows the output of the script: ['Senin', 'Selasa', 'Rabu', 'Kamis'].

4. Tanda Kutip

Digunakan untuk menandai string dimana tanda kutip tiga digunakan untuk string multibaris.

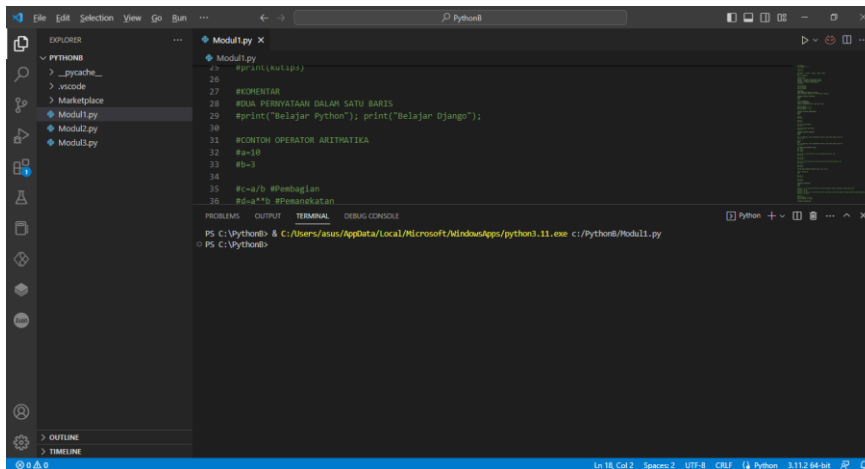
Contoh program penggunaan tanda kutip:

```
16
17 #TANDA KUTIP
18 kutip1 = 'Belajar Pemrograman python'
19 kutip2 = "Belajar pemrograman python"
20 kutip3 = '''Belajar pemrograman
21 python'''
22
23 print(kutip1)
24 print(kutip2)
25 print(kutip3)
```

The terminal shows the output of the script: Belajar Pemrograman python, Belajar pemrograman python, Belajar pemrograman python.

5. Komentar

Tanda pagar (#) atau komentar digunakan untuk menandai komentar di python. Komentar tidak akan diproses oleh interpreter Python.



```
Modul1.py
#Komentar
#Dua Pernyataan Dalam Satu Baris
print("Belajar Python"); print("Belajar Django");
#Contoh Operator Aritmatika
#1+2
#3-1
#4/2
#5**2
#Penugasan
```

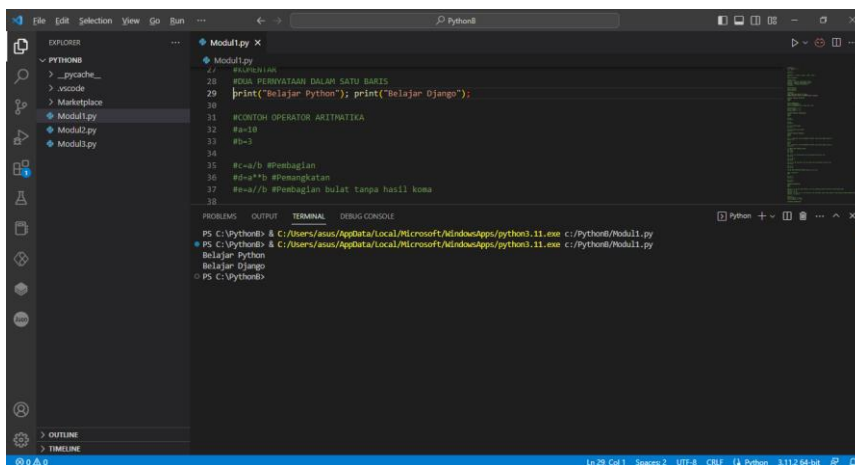
PS C:\Python> python Modul1.py

Belajar Python
Belajar Django

6. Dua Pernyataan Dalam Satu Baris

Titik koma dapat digunakan ketika terdapat 2 pernyataan dalam 1 baris kode.

Contoh program dua pernyataan dalam satu baris :



```
Modul1.py
#Komentar
#Dua Pernyataan Dalam Satu Baris
print("Belajar Python"); print("Belajar Django");
#Contoh Operator Aritmatika
#1+2
#3-1
#4/2
#5**2
#Penugasan
```

PS C:\Python> python Modul1.py

Belajar Python
Belajar Django

Tipe Data

Macam tipe data antara lain:

- int, tipe data yang dapat Anda isi dengan bilangan bulat.
- float, tipe data yang dapat Anda isi dengan bilangan koma.
- string, tipe data yang dapat Anda isi dengan sebuah karakter atau kalimat.
- complex, tipe data bilangan kompleks atau bilangan imajiner, seperti 5j, 54j, 1j.
- long, tipe data yang dapat diisi dengan bilangan yang sangat besar.
- boolean, tipe data yang nilainya hanya True dan False.
- List, adalah tipe data yang berisi item yang berurut.
- Tuple, adalah jenis data lain yang mirip dengan list. Perbedaannya dengan list adalah anggotanya tidak bisa diubah (immutable). Tuple dideklarasikan dengan menggunakan tanda kurung (). dan anggotanya dipisahkan oleh tanda koma.
- Set, adalah salah satu tipe data yang tidak berurut (unordered).
- Dictionary, adalah tipe data yang tiap anggotanya terdiri dari pasangan kuncinilai (key-value). Dictionary dideklarasikan dengan menggunakan tanda kurung kurawal { }, dimana anggotanya memiliki bentuk kunci:nilai atau key:value dan tiap anggota dipisah tanda koma. Kunci dan nilainya bisa memiliki tipe sembarang.

1. Operator aritmatika

Digunakan untuk melakukan operasi matematika, seperti penjumlahan, pengurangan, perkalian, pembagian, dan sebagainya.

Contoh Kode Operator Aritmatika

```
30
31 #CONTOH OPERATOR ARITMATIKA
32 a=10
33 b=3
34
35 c=a/b #Pembagian
36 d=a*b #Pangkat
37 e=a//b #Pembagian bulat tanpa hasil koma
38
39 print("Hasil :", c)
40 print("Pangkat :", d)
41 print("Hasil :", e)
42
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python\Modul1.py
Hasil : 3.3333333333333335
Pangkat : 1000
Hasil : 3
PS C:\Python>
```

2. Operator Perbandingan

Digunakan untuk membandingkan 2 buah nilai. Hasil perbandingannya antara True atau False tergantung kondisi.

Contoh program operator perbandingan :

```
43 #CONTOH OPERATOR PERBANDINGAN
44 a=10
45 b=1
46
47 c=a<b
48 print(c)
49
50 c=a>b
51 print(c)
52
53 c=a<=b
54 print(c)
55
56 c=a>=b
57 print(c)
58
59 c=a==b
60 print(c)
61
62 c=a!=b
63 print(c)
64
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python\Modul1.py
False
True
False
True
False
True
PS C:\Python>
```

3. Operator Penugasan

Digunakan untuk memberi nilai ke variabel. Contoh program kode operator penugasan

```
58
59 # CONTOH OPERATOR PEMASANGAN
60 a=10
61 b=3
62
63 a += b #Operasi untuk menambahkan operator yang kanan dengan yang kiri
64 print(a)
65
66 a-=b
67 b-=2
68 a -= b #Operasi untuk menambahkan operator yang kanan dengan yang kiri
69 print(a)
70
71
72
```

```
PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python6/Modul1.py
13
7
30
3.3333333333333335
PS C:\Python>
```

4. Operator Logika

Digunakan untuk melakukan operasi logika. Contoh program kode operator logika :

```
73
74 # CONTOH CODE OPERATOR LOGIKA
75 a= True
76 b= False
77 c= True
78 d= a and c # and hasilnya true jika keduanya bernilai true
79 print(d)
80 d= a not b
81 print(d)
82 d= a or b # or hasilnya true jika salah satu operatonya bernilai true
83 print(d)
84 d= a == c
85 print(d)
86
87
```

```
PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python6/Modul1.py
True
False
True
True
PS C:\Python>
```

5. Operator Bitwise

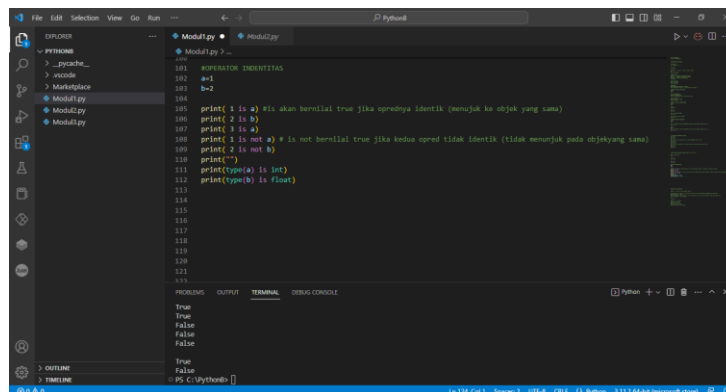
Operator yang melakukan operasi bit terhadap operand, operator ini beroperasi bit per bit. Contoh program kode operator bitwise :

```
88
89 # CONTOH KODE OPERATOR BITWISE operasi bit per bit
90
91 a=1 # binary 10
92 b=2
93
94 c= a | b
95 print(c)
96
97 c= a & b
98 print(c)
99
100
```

```
PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python6/Modul1.py
3
0
3
PS C:\Python>
```

6. Operator Identitas

Adalah operator yang memeriksa apakah dua buah nilai (atau variabel) berada pada lokasi memori yang sama. Contoh program kode operator identitas:

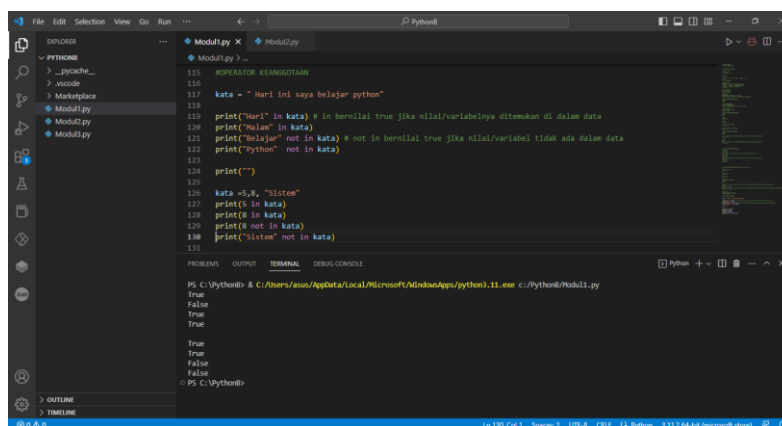


```
101 OPERATOR IDENTITAS
102 a=1
103 b=1
104
105 print(1 is a) #is akan bernilai true jika operanya identik (menunjuk ke objek yang sama)
106 print(2 is b)
107 print(3 is a)
108 print(2 is not a) # is not bernilai true jika kedua operd tidak identik (tidak menunjuk pada objek yang sama)
109 print(2 is not b)
110 print("")
111 print(type(a) is int)
112 print(type(b) is float)
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

7. Operator Keanggotaan

Digunakan untuk memeriksa apakah suatu nilai atau variabel merupakan anggota atau ditemukan di dalam suatu data (string, list, tuple, set, dan dictionary).

Contoh program kode operator keanggotaan :



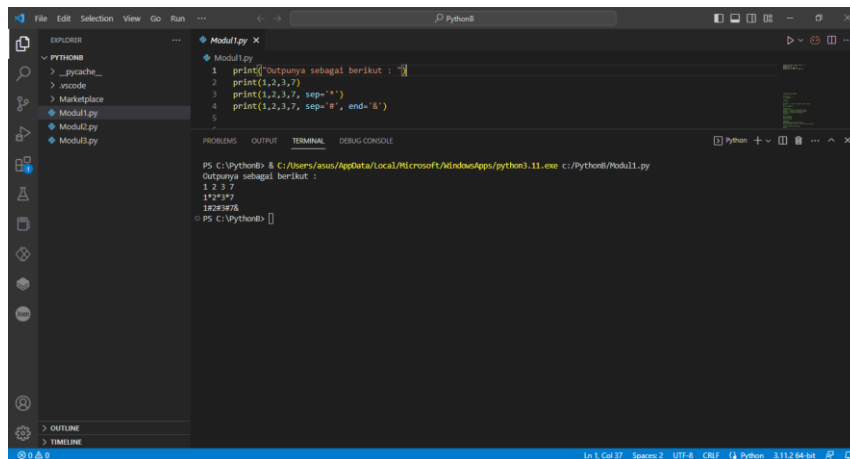
```
115 OPERATOR KEANGGOTAAN
116
117 kata = " Hari ini saya belajar python"
118
119 print("Hari" in kata) # in bernilai true jika nilai/variabelnya ditemukan di dalam data
120 print("Nulan" in kata)
121 print("belajar" not in kata) # not in bernilai true jika nilai/variabel tidak ada dalam data
122 print("python" not in kata)
123
124 print("")
125
126 kata = "Sis", "Sisitem"
127 print("S" in kata)
128 print("S" in kata)
129 print("S" not in kata)
130 print("Sisitem" not in kata)
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

MODUL 2

1. Input dan Output

a. Output

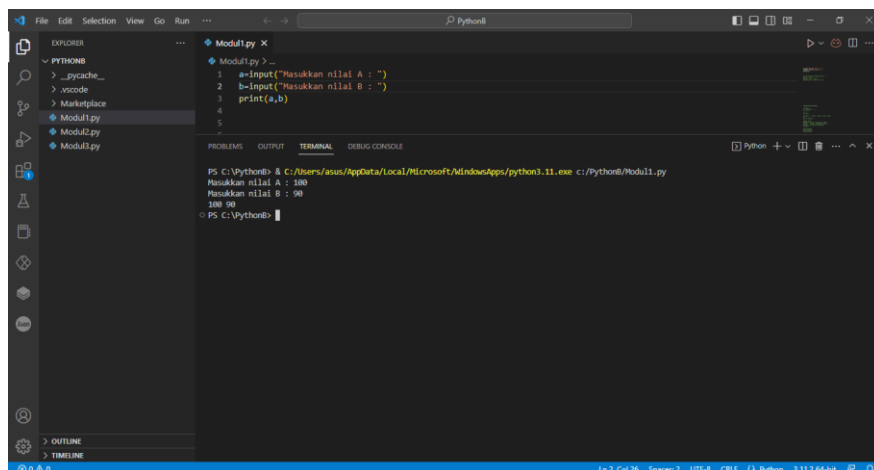
Disini terdapat fungsi output dimana fungsi yang biasa digunakan adalah print(). Print sendiri memiliki fungsi untuk menampilkan hasil atau output dari program yang kita buat. Dimana terdapat sep berfungsi sebagai tanda pemisah antar objek yang dicetak. end merupakan karakter yang akan dicetak di akhir baris. file adalah nama file kemana objek akan dicetak. Contoh program kode output :



```
File Edit Selection View Go Run ... Python8
EXPLORER
PYTHON
  __pycache__
  __vscode__
  Marketplace
  Modul1.py
  Modul2.py
  Modul3.py
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
Modul1.py X
1 print("Outputnya sebagai berikut : ")
2 print(1,2,3,7)
3 print(1,2,3,7, sep=" ")
4 print(1,2,3,7, sep=" ", end="& ")
5
PS C:\Python8 & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/Python8/Modul1.py
Outputnya sebagai berikut :
1 2 3 7
1 2 3 7
1&2&3&7&
PS C:\Python8 >
```

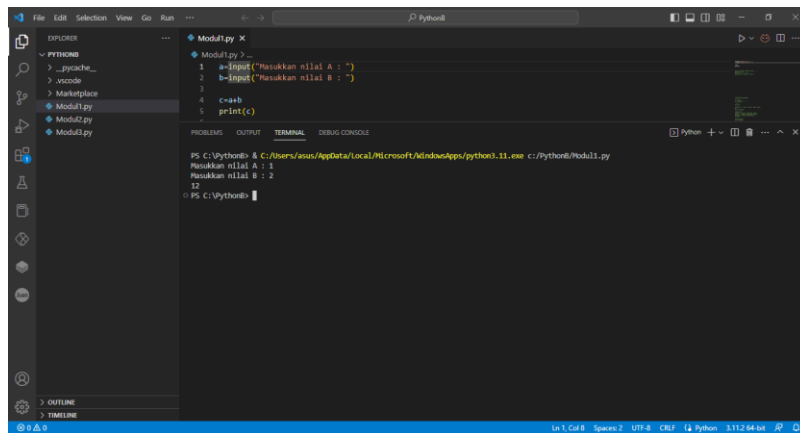
b. Input

Fungsi input ini biasanya digunakan adalah input() untuk mengimput masukan dari program yang kita buat. Input, proses, dan output adalah inti dari semua program komputer. Contoh program kode input :



```
File Edit Selection View Go Run ... Python
EXPLORER
PYTHON
  __pycache__
  __vscode__
  Marketplace
  Modul1.py
  Modul2.py
  Modul3.py
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
Python
Modul1.py > ...
1 a=input("Masukkan nilai A : ")
2 b=input("Masukkan nilai B : ")
3 print(a,b)
4
5
PS C:\Python8 & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/Python8/Modul1.py
Masukkan nilai A : 100
Masukkan nilai B : 90
100 90
PS C:\Python8 >
```

Contoh program kode input tanpa menggunakan fungsi int() :



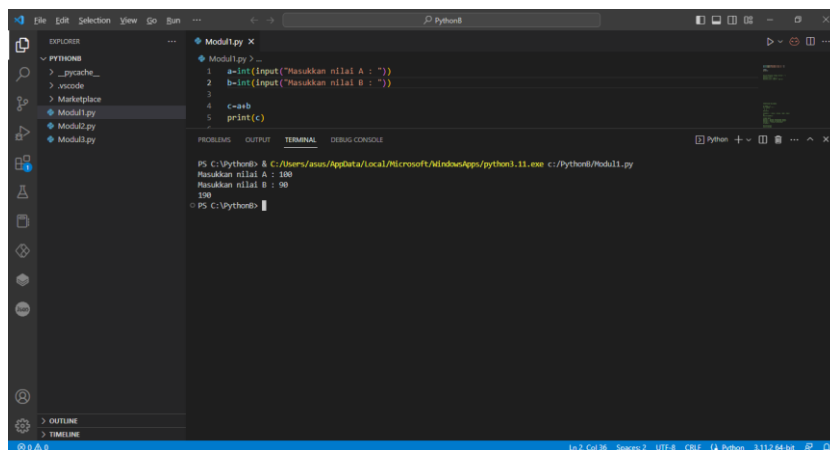
The screenshot shows the Visual Studio Code editor with a Python file named 'Modul1.py'. The code is as follows:

```
1 a=input("Masukkan nilai A : ")
2 b=input("Masukkan nilai B : ")
3
4 c=a+b
5 print(c)
```

The terminal output shows the execution of the script:

```
PS C:\Python> & C:\Users\assu\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python\Modul1.py
Masukkan nilai A : 1
Masukkan nilai B : 2
3
```

Contoh program kode input pertama menggunakan fungsi int() :



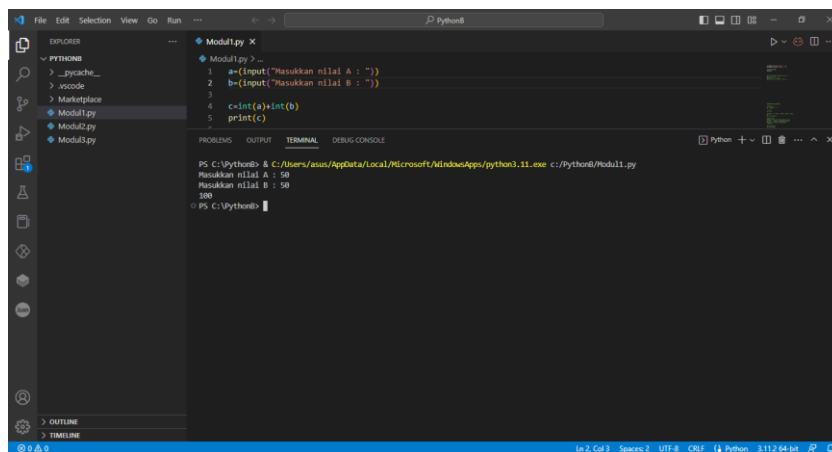
The screenshot shows the Visual Studio Code editor with a Python file named 'Modul1.py'. The code is as follows:

```
1 a=int(input("Masukkan nilai A : "))
2 b=int(input("Masukkan nilai B : "))
3
4 c=a+b
5 print(c)
```

The terminal output shows the execution of the script:

```
PS C:\Python> & C:\Users\assu\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python\Modul1.py
Masukkan nilai A : 100
Masukkan nilai B : 90
190
```

Contoh program kode input kedua menggunakan fungsi int() :



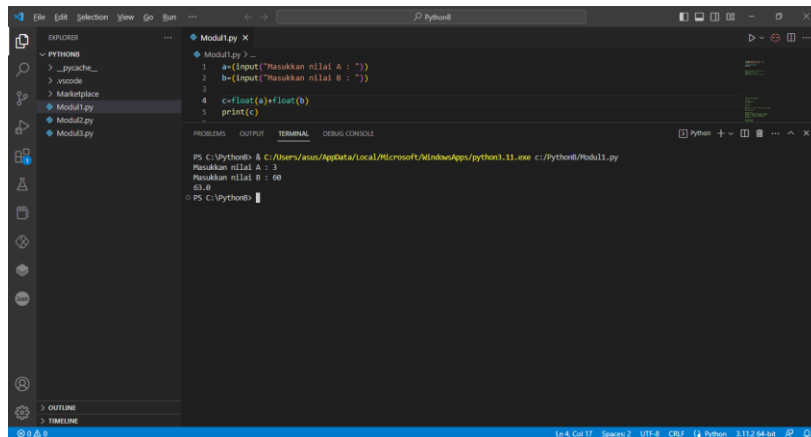
The screenshot shows the Visual Studio Code editor with a Python file named 'Modul3.py'. The code is as follows:

```
1 a=input("Masukkan nilai A : ")
2 b=input("Masukkan nilai B : ")
3
4 c=int(a)*int(b)
5 print(c)
```

The terminal output shows the execution of the script:

```
PS C:\Python> & C:\Users\assu\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python\Modul3.py
Masukkan nilai A : 50
Masukkan nilai B : 50
2500
```

Contoh program kode input menggunakan fungsi float() untuk hasil pecahan :



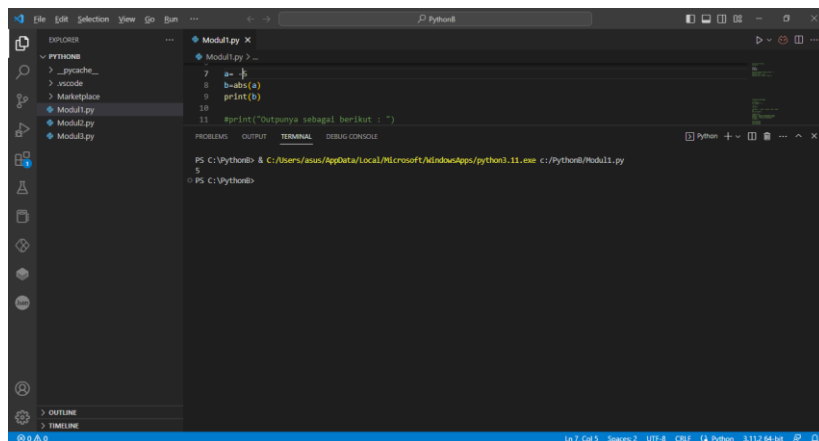
The screenshot shows a VS Code editor with a Python file named 'Modul1.py'. The code is as follows:

```
1 a=input("Masukkan nilai A : ")
2 b=input("Masukkan nilai B : ")
3
4 c=float(a)+float(b)
5 print(c)
```

The terminal output shows the program being executed with inputs 3 and 60, resulting in 63.0.

```
PS C:\Python> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python/Modul1.py
Masukkan nilai A : 3
Masukkan nilai B : 60
63.0
PS C:\Python>
```

Contoh program kode input menggunakan fungsi abs() dinamis untuk menghilangkan tipe data yang ada minusnya :



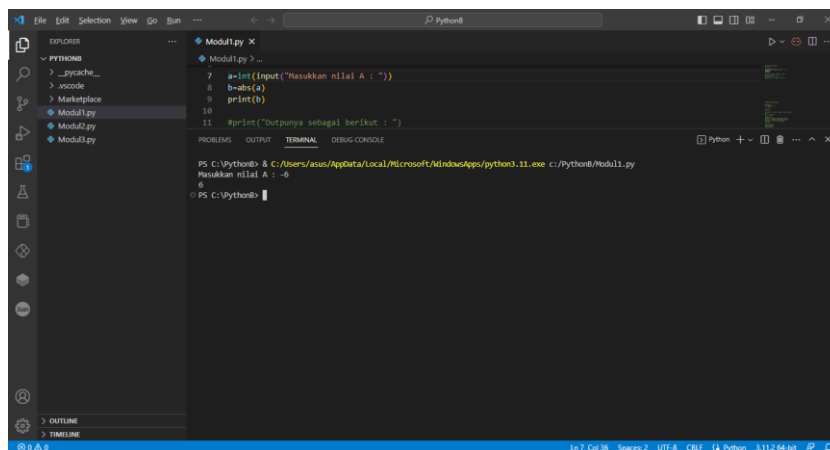
The screenshot shows a VS Code editor with a Python file named 'Modul1.py'. The code is as follows:

```
7 a= float(input("Masukkan nilai A : "))
8 b=abs(a)
9 print(b)
10
11 #print("Outputnya sebagai berikut : ")
```

The terminal output shows the program being executed with input 5, resulting in 5.

```
PS C:\Python> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python/Modul1.py
5
PS C:\Python>
```

Contoh program kode input menggunakan fungsi abs() statis untuk menghilangkan tipe data yang ada minusnya :



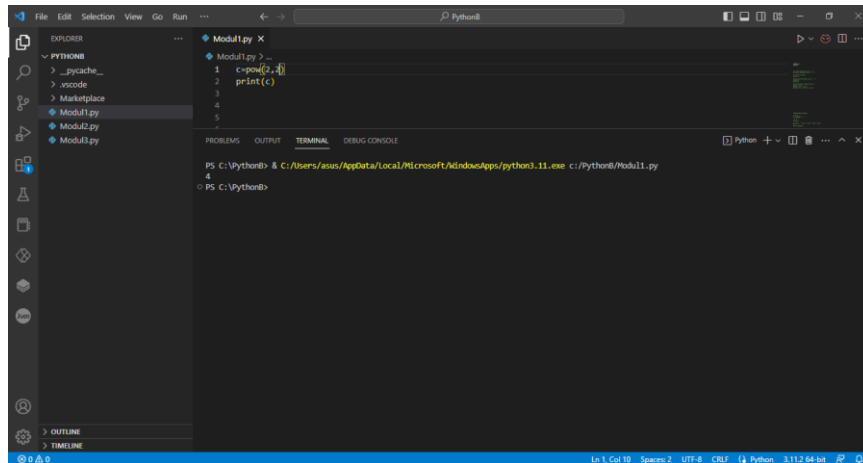
The screenshot shows a VS Code editor with a Python file named 'Modul1.py'. The code is as follows:

```
7 a= float(input("Masukkan nilai A : "))
8 b=abs(a)
9 print(b)
10
11 #print("Outputnya sebagai berikut : ")
```

The terminal output shows the program being executed with input -6, resulting in 6.

```
PS C:\Python> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python/Modul1.py
Masukkan nilai A : -6
6
PS C:\Python>
```

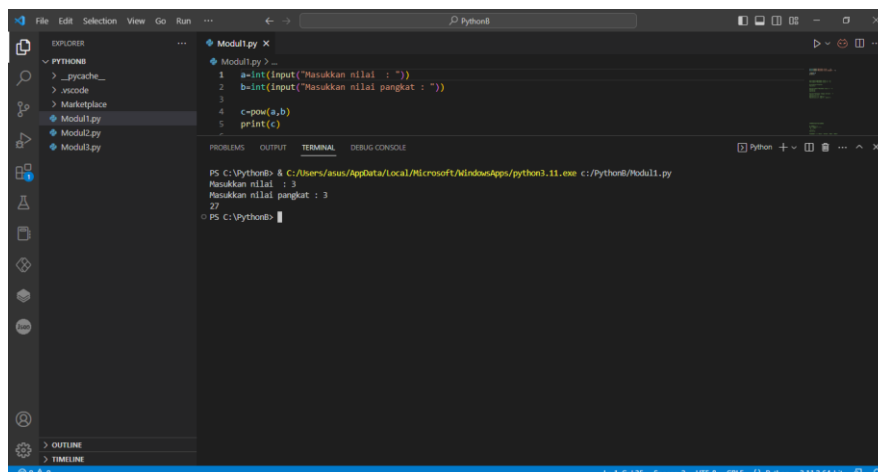

Contoh program kode input menggunakan fungsi pow() Statis untuk menghitung pangkat :



```
Modul1.py
1 c=pow(2,4)
2 print(c)
3
4
5
```

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul1.py
4
PS C:\Python8>
```

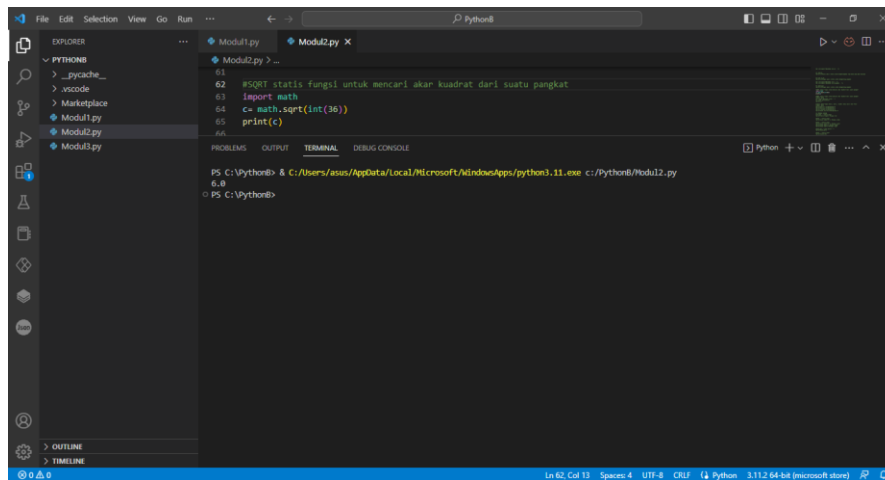
Contoh program kode input menggunakan fungsi pow() Statis untuk menghitung pangkat :



```
Modul1.py
1 a=int(input("Masukkan nilai : "))
2 b=int(input("Masukkan nilai pangkat : "))
3
4 c=pow(a,b)
5 print(c)
6
```

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul1.py
Masukkan nilai : 3
Masukkan nilai pangkat : 3
27
PS C:\Python8>
```

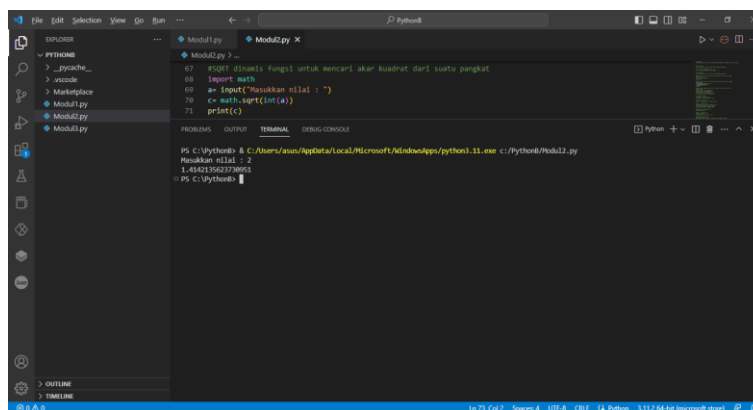
Contoh program kode input menggunakan sqrt() statis. Fungsi untuk mencari akar kuadrat dari suatu pangkat.



```
61
62 #SQRT statis fungsi untuk mencari akar kuadrat dari suatu pangkat
63 import math
64 c= math.sqrt(int(36))
65 print(c)
66
```

```
PS C:\Python8> & C:/Users/assu/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul2.py
6.0
PS C:\Python8>
```

Contoh program kode input menggunakan sqrt() dinamis. Fungsi untuk mencari akar kuadrat dari suatu pangkat.

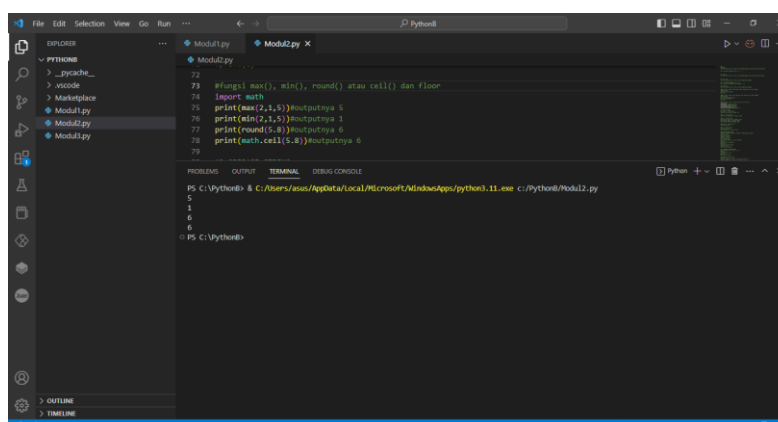


```
67 #SQRT dinamis fungsi untuk mencari akar kuadrat dari suatu pangkat
68 import math
69 a= input("Masukkan nilai : ")
70 c= math.sqrt(int(a))
71 print(c)

```

```
PS C:\Python8> & C:/Users/assu/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul2.py
Masukkan nilai : 2
1.4142135623730951
PS C:\Python8>
```

Contoh lain program kode input menggunakan sqrt(). fungsi lain seperti max(), untuk menampilkan nilai paling akhir. min(), menampilkan nilai paling awal. round() atau ceil() untuk pembulatan keatas. Dan floor() untuk pembulatan kebawah.

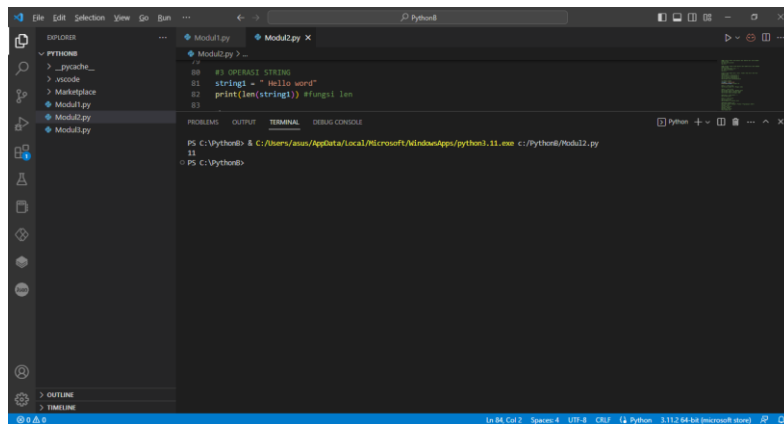


```
72
73 #fungsi max(), min(), round() atau ceil() dan floor
74 import math
75 print(max(2,1,5))#outputnya 5
76 print(min(2,1,5))#outputnya 1
77 print(round(5.4))#outputnya 5
78 print(math.ceil(5.4))#outputnya 6
79
```

```
PS C:\Python8> & C:/Users/assu/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul2.py
5
1
6
6
PS C:\Python8>
```

c. Operasi String

len() berfungsi untuk mengembalikan panjang (jumlah anggota) dari suatu objek. Contoh program fungsi len():

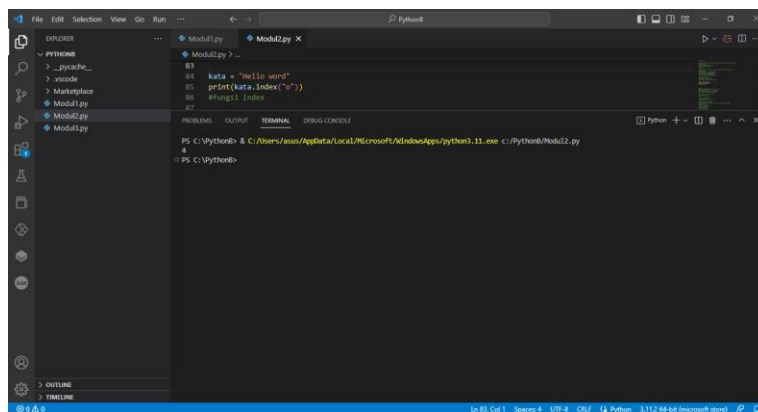


```
10 # OPENING STRING
11 string = " Hello word"
12 print(len(string)) #fungsi len
13
```

Terminal output:

```
PS C:\Python> & C:/Users/assu/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python/Modul2.py
11
PS C:\Python>
```

Fungsi index() untuk mencari posisi suatu nilai. Contoh program fungsi index():



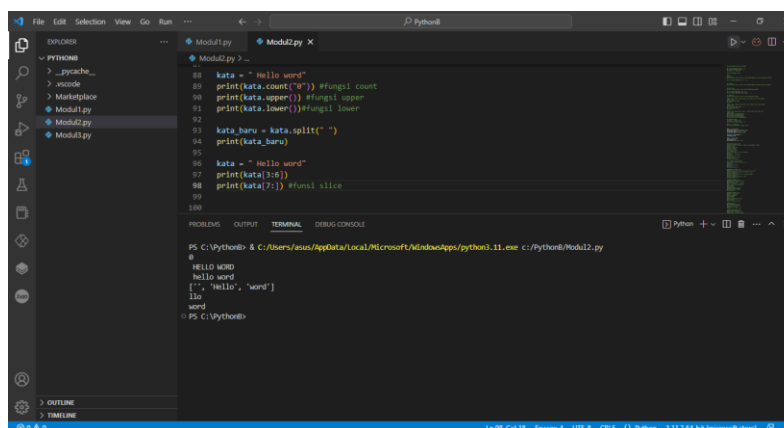
```
83
84 kata = "Hello word"
85 print(kata.index("o")) #fungsi index
86
```

Terminal output:

```
PS C:\Python> & C:/Users/assu/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python/Modul2.py
8
PS C:\Python>
```

Fungsi lain juga terdiri dari count() untuk menghitung kemunculan nilai tertentu. upper() untuk mengubah string menjadi huruf kapital. lower() untuk mengubah string menjadi huruf kecil. split() untuk memisah string menjadi item.

Contoh program:



```
88 kata = " Hello word"
89 print(kata.count("o")) #fungsi count
90 print(kata.upper()) #fungsi upper
91 print(kata.lower()) #fungsi lower
92
93 kata_baru = kata.split(" ")
94 print(kata_baru)
95
96 kata = " Hello word"
97 print(kata[1:6])
98 print(kata[2:]) #fungsi slice
99
100
```

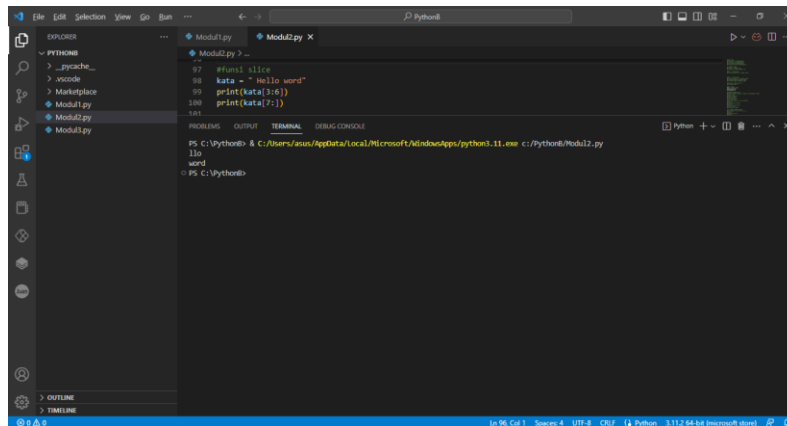
Terminal output:

```
PS C:\Python> & C:/Users/assu/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python/Modul2.py
#
HELLO WORD
hello word
['', 'hello', 'word']
110
word
PS C:\Python>
```

d. Range Slice.

Untuk menampilkan range karakter dari a mendekati b (limit b), yang diformulasikan dengan.

Contoh program range slice:



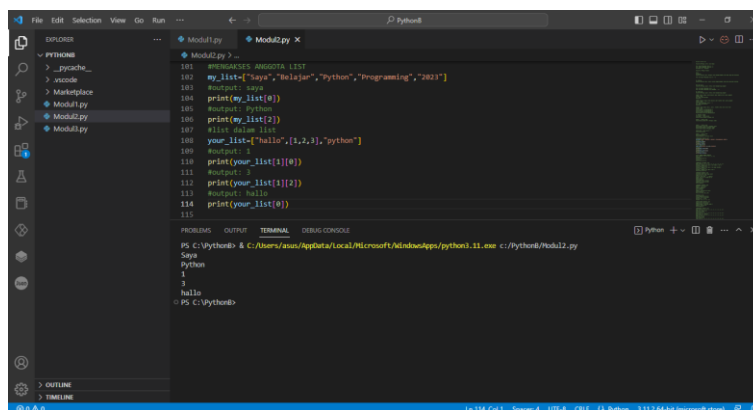
```
17 #range slice
18 kata = " Hello word"
19 print(kata[3:6])
20 print(kata[7:11])
```

Terminal output:

```
PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps/python3.11.exe c:\python\Modul2.py
19
20
word
PS C:\Python>
```

e. List

Salah satu tipe data built-in Python, yang dapat digunakan kapan saja tanpa harus meng-import modul terlebih dahulu. Contoh program mengakses anggota list. Kita bisa mengakses anggota list dengan menggunakan indeksnya dengan format `namalist[indeks]` :



```
181 #membuat list
182 my_list = ["saya", "belajar", "python", "programming", "2023"]
183 #output: saya
184 print(my_list[0])
185 #output: python
186 print(my_list[2])
187 #list dalam list
188 your_list = ["hallo", [1,2,3], "python"]
189 #output: 1
190 print(your_list[1][0])
191 #output: 1
192 print(your_list[1][2])
193 #output: hallo
194 print(your_list[0])
195
```

Terminal output:

```
PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps/python3.11.exe c:\python\Modul2.py
Saya
Python
1
3
hallo
PS C:\Python>
```

List Dengan Indeks Negatif Python mendukung indeks negatif, yaitu urutan dimulai dari anggota terakhir. Indeks anggota paling belakang adalah -1, kemudian -2, dan seterusnya. Contoh list dengan index negatif :

```
115 # Mengubah anggota list dengan slicing
116 my_list = ['p', 'y', 't', 'h', 'o', 'n']
117 print(my_list[1])
118 # Output: y
119 print(my_list[1:])
120 # Output: ['p', 'y', 't', 'h', 'o', 'n']
121 print(my_list[:3])
122 # Output: ['p', 'y', 't']
123 # Mengubah anggota list dengan slicing
124 my_list = ['p', 'y', 't', 'h', 'o', 'n']
125 # Output: anggota list dari 1 sd 5 (dari 0 sd 4)
126 print(my_list[1:5])
127 # Output: anggota list dari 1 sd 4 yang terakhir
128 print(my_list[1:4])
129 # Output: anggota list dari 0 sd 4
```

Memotong (Slicing) List dengan mengakses anggota list dari range tertentu dengan menggunakan operator slicing titik dua (:). Contoh memotong list:

```
121 # Mengubah anggota list dengan slicing
122 my_list = ['p', 'y', 't', 'h', 'o', 'n']
123 print(my_list[1:4])
124 # Output: anggota list dari 1 sd 4 yang terakhir
125 print(my_list[4:])
126 # Output: anggota list dari 4 sd 4
127 print(my_list[:5])
```

-Mengubah Anggota List merupakan tipe data yang bersifat mutable, artinya anggotanya bisa diubah. Contoh program mengubah anggota list :

```
132 # Mengubah anggota list
133 # Misalnya ada nilai yang salah
134 ganjil = [1, 3, 4, 7, 9]
135 print("list awal : ", ganjil)
136 # Ubah nilai ke 3 (index ke 2)
137 ganjil[2] = 5
138 print(ganjil)
139 # Mengubah sekali banyak
140 ganjil[2:5] = [11, 13, 15]
141 print(ganjil)
```

-Menambahkan Anggota List

Fungsi `append()` berguna untuk menambahkan anggota ke dalam list. Selain itu, ada metode `extend()` untuk menambahkan anggota list ke

dalam list. Kita juga bisa menggunakan operator + untuk menggabungkan dua list, dan operator * untuk melipatgandakan list. Contoh program Menambah Anggota List :

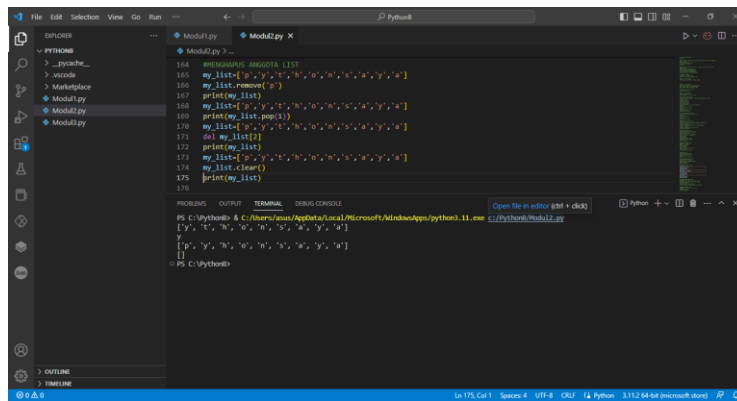
```
142  
143 MENAMBAH ANGGOTA LIST  
144 ganjil = [1,3,5,7]  
145 ganjil.append(9)  
146 print(ganjil)  
147 [1, 3, 5, 7, 9]  
148 ganjil.extend([11,13,15])  
149 print(ganjil)  
150 [1, 3, 5, 7, 9, 11, 13, 15]  
151  
152 genap=[2,4,6]  
153 print(genap+[8,10,12])  
154 Output [2, 4, 6, 8, 10, 12]  
155 print(' '.join(str(x) for x in ganjil))  
156 Output [1, 3, 5, 7, 9, 11, 13, 15]  
157
```

-Contoh program menyisipkan anggota lis:

```
158 MENYISIPKAN ANGGOTA LIST  
159 ganjil=[5,7,11,13,15]  
160 Menyisipkan 9 setelah angka 7  
161 ganjil.insert(2,9)  
162 print(ganjil)  
163
```

-Menghapus Anggota List

Dengan menggunakan metode remove(), pop(), atau kata kunci del untuk menghapus anggota list. Selain itu kita bisa menggunakan clear() untuk mengosongkan list. Fungsi pop() selain menghapus anggota list, juga mengembalikan nilai indeks anggota tersebut. Contoh program menghapus anggota list :



```
154 MENGENALAN ANGGOTA LIST
155 my_list=['p','y','t','h','o','n','s','a','y','a']
156 my_list.remove('p')
157 print(my_list)
158 my_list=['p','y','t','h','o','n','s','a','y','a']
159 print(my_list.pop())
160 my_list=['p','y','t','h','o','n','s','a','y','a']
161 del my_list[2]
162 print(my_list)
163 my_list=['p','y','t','h','o','n','s','a','y','a']
164 my_list.clear()
165 print(my_list)
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Python> & C:\Users\user\AppData\Local\Microsoft\WindowsApps\python.11.exe c:/Python/Modul2.py

['y', 't', 'h', 'o', 'n', 's', 'a', 'y', 'a']

y

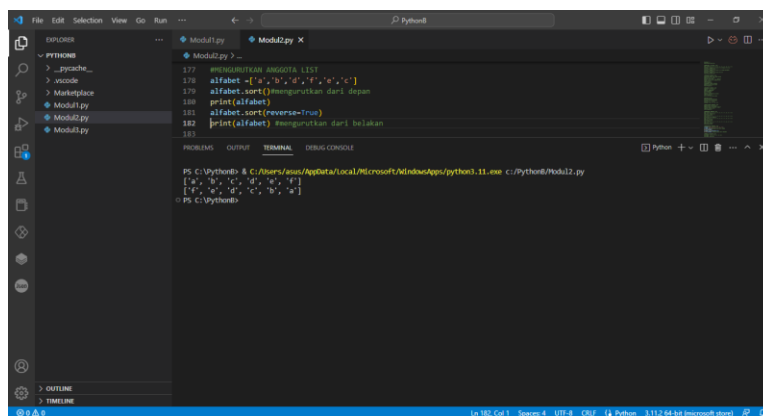
['p', 'y', 't', 'h', 'o', 'n', 's', 'a', 'y', 'a']

[]

PS C:\Python>

-Mengurutkan Anggota List

Untuk mengurutkan atau menyortir anggota list, kita bisa menggunakan metode `sort()`. Untuk membalik dengan urutan sebaliknya bisa dengan menggunakan argumen. Contoh program mengurutkan anggota list :



```
177 MENGENALAN ANGGOTA LIST
178 alfabet=['a','b','d','f','e','c']
179 alfabet.sort() #mengurutkan dari depan
180 print(alfabet)
181 alfabet.sort(reverse=True)
182 print(alfabet) #mengurutkan dari belakang
183
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

PS C:\Python> & C:\Users\user\AppData\Local\Microsoft\WindowsApps\python.11.exe c:/Python/Modul2.py

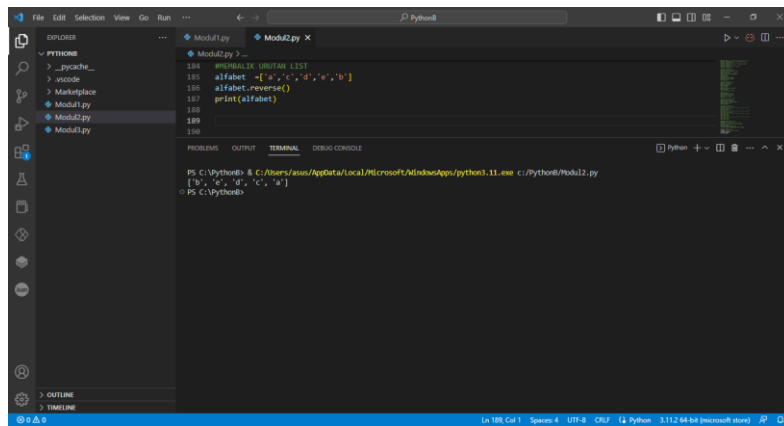
['a', 'b', 'c', 'd', 'e', 'f']

['f', 'e', 'd', 'c', 'b', 'a']

PS C:\Python>

-Membalik Urutan List

Selain mengurutkan, kita juga bisa membalikkan urutan list dengan menggunakan metode `reverse()`. Contoh program membalik urutan list:

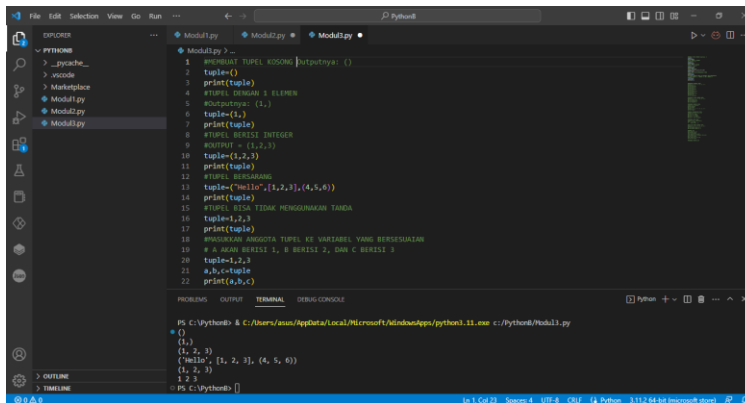


MODUL 3

1. Membuat Tuple

Tuple mirip dengan list dibuat dengan meletakkan semua anggota di dalam tanda kurung (), masing-masing dipisahkan oleh tanda koma.

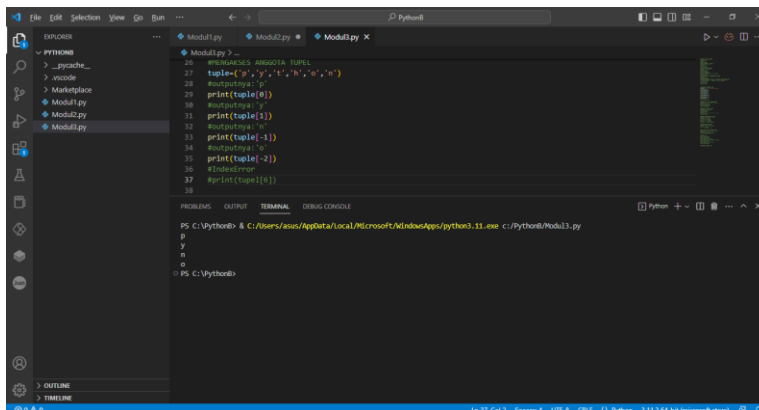
Contoh program membuat tuple :



```
1 # Membuat Tuple
2 tuple_kosong = tuple()
3 print(tuple_kosong)
4 # Tuple dengan 3 elemen
5 outputnya = ('1', '2', '3')
6 print(outputnya)
7 # Tuple dengan 1 elemen
8 outputnya = ('1',)
9 print(outputnya)
10 # Tuple dengan 2 elemen
11 outputnya = ('1', '2')
12 print(outputnya)
13 # Tuple dengan 4 elemen
14 outputnya = ('1', '2', '3', '4')
15 print(outputnya)
16 # Tuple dengan 1 elemen yang bernomor
17 outputnya = ('1',)
18 print(outputnya)
19 # Tuple dengan 2 elemen yang bernomor
20 outputnya = ('1', '2')
21 print(outputnya)
22 # Tuple dengan 3 elemen yang bernomor
23 outputnya = ('1', '2', '3')
24 print(outputnya)
```

2. Mengakses Anggota Tuple

Seperti halnya list, tuple bisa mengakses anggota tuple lewat indeksinya menggunakan format `namatuple[indeks]`. Contoh program mengakses tuple :



```
1 # Mengakses Anggota Tuple
2 tuple_kosong = tuple()
3 print(tuple_kosong)
4 # Tuple dengan 3 elemen
5 outputnya = ('1', '2', '3')
6 print(outputnya)
7 # Mengakses anggota tuple
8 outputnya = ('1', '2', '3')
9 print(outputnya[0])
10 print(outputnya[1])
11 print(outputnya[2])
12 # Mengakses anggota tuple dengan range
13 outputnya = ('1', '2', '3')
14 print(outputnya[0:2])
15 print(outputnya[1:3])
16 print(outputnya[0:3])
17 print(outputnya[0:4])
```

Sama seperti list, kita bisa mengakses satu range anggota tuple dengan menggunakan operator titik dua (:). Contoh program mengakses tuple dengan range :

```
39 #Membuat tuple dengan range
40 tuple1 = 'a', 'b', 'c', 'd', 'e'
41 print(tuple1)
42 print(tuple(2:4))
43 print(tuple(3:))
44
```

Output:

```
PS C:\Python> & C:/Users/ansu/AppData/Local/Microsoft/WindowsApps/python.11.exe c:/Python/Modul3.py
('a', 'b', 'c', 'd', 'e')
('b', 'c', 'd')
('c', 'd', 'e')
```

3. Mengubah Anggota Tuple

Setelah tuple dibuat, maka anggota tuple tidak bisa lagi diubah atau dihapus. Akan tetapi, bila anggota tuple-nya adalah tuple bersarang dengan anggota seperti list, maka item pada list tersebut dapat diubah. Contoh program mengubah anggota tuple :

```
45 #Membuat anggota tuple
46 tuple2 = ('a', 'b', 'c', 'd', ('e', 'f'))
47 tuple2[4][0] = 'g'
48 print(tuple2)
49 #Membuat tuple dengan list
50 print(tuple2)
51
```

Output:

```
PS C:\Python> & C:/Users/ansu/AppData/Local/Microsoft/WindowsApps/python.11.exe c:/Python/Modul3.py
('a', 'b', 'c', 'd', ('e', 'f'))
('a', 'b', 'c', 'd', ('g', 'f'))
```

4. Menguji Keanggotaan Tuple

Seperti halnya string dan list, kita bisa menguji apakah sebuah objek adalah anggota dari tuple atau tidak, yaitu dengan menggunakan operator in atau not in untuk kebalikannya. Contoh program menguji anggota tuple :

```
52 #Membuat keanggotaan tuple
53 tuple1 = ('a', 'b', 'c', 'd', 'e')
54 print('a' in tuple1)
55 print('f' in tuple1)
56 print('a' not in tuple1)
57 print('f' not in tuple1)
58
```

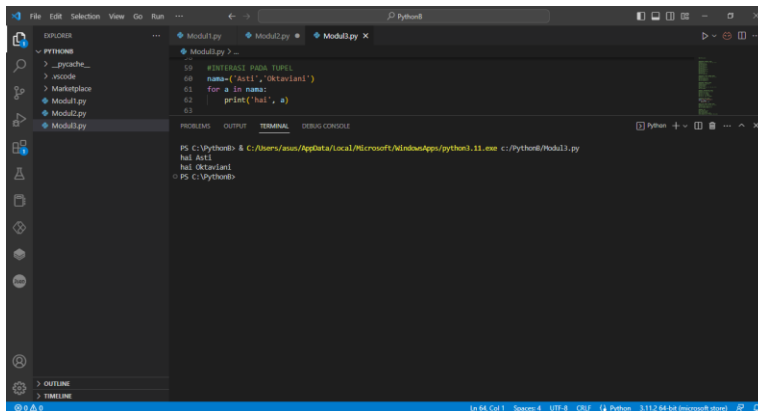
Output:

```
PS C:\Python> & C:/Users/ansu/AppData/Local/Microsoft/WindowsApps/python.11.exe c:/Python/Modul3.py
True
False
True
False
```

5. Iterasi pada Tuple

Kita bisa menggunakan for untuk melakukan iterasi pada tiap anggota dalam tuple.

Contoh program iterasi pada tuple :



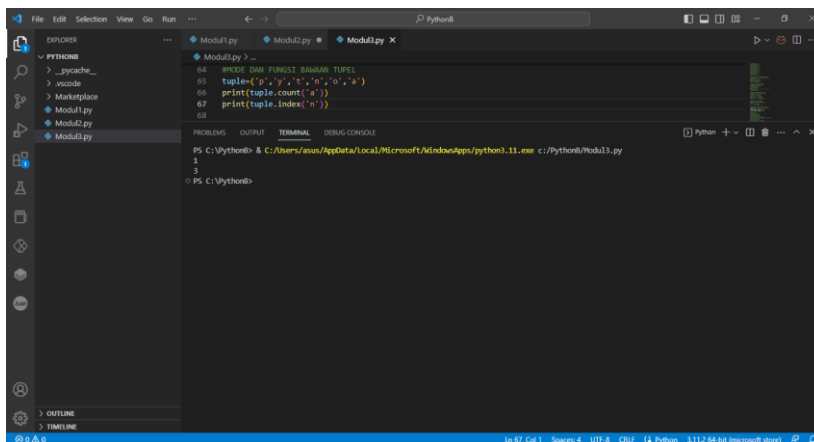
```
59 #ITERASI PADA TUPLE
60 nama=('Asli','Oktaiani')
61 for a in nama:
62     print('hai', a)
63
```

Terminal output:

```
PS C:\Python> & C:/Users/ausu/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python/Modul3.py
hai Asli
hai Oktaiani
PS C:\Python>
```

6. Metode dan Fungsi Bawaan Tuple

Tuple hanya memiliki dua buah metode yaitu count() dan index(). Metode count(x) berfungsi mengembalikan jumlah item yang sesuai dengan x pada tuple. Metode index(x) berfungsi mengembalikan indeks dari item pertama yang sama dengan x. Contoh program fungsi bawaan tuple :



```
64 #METODE DAN FUNGSI BAWAAN TUPLE
65 tuple=('p','y','t','h','o','n')
66 print(tuple.count('p'))
67 print(tuple.index('n'))
68
```

Terminal output:

```
PS C:\Python> & C:/Users/ausu/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python/Modul3.py
1
5
PS C:\Python>
```

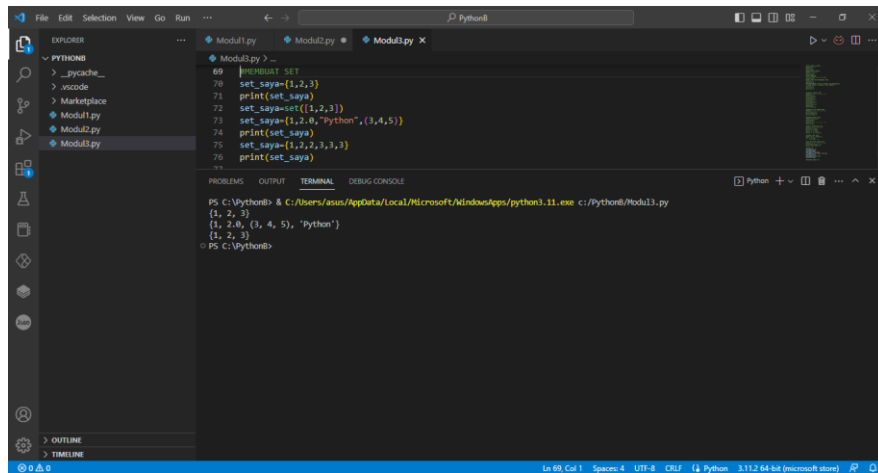
Tipe Data Set

Set adalah salah satu tipe data yang tidak berurut (unordered). Set memiliki anggota yang unik (tidak ada duplikasi).

1. Membuat Set

Set dibuat dengan meletakkan anggota - anggotanya di dalam tanda kurung kurawal {}, dipisahkan menggunakan tanda koma. Kita juga bisa membuat set dari list dengan memasukkan list ke dalam fungsi set() Set bisa berisi data

campuran, baik integer, float, string, dan lain sebagainya. Akan tetapi set tidak bisa berisi list, set, dan dictionary. Contoh program membuat set :



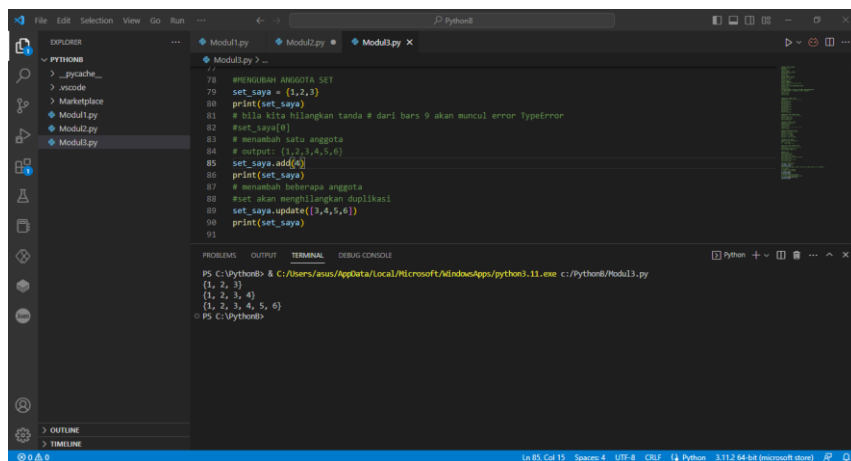
```
69 #membuat SET
70 set_saya={1,2,3}
71 print(set_saya)
72 set_saya=set([1,2,3])
73 set_saya={1,2,0,"Python",(3,4,5)}
74 print(set_saya)
75 set_saya={1,2,2,3,3}
76 print(set_saya)
```

Terminal Output:

```
PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python\Modul3.py
{1, 2, 3}
{1, 2, 0, (3, 4, 5), 'Python'}
{1, 2, 3}
```

2. Mengubah Anggota Set

Set bersifat mutable. Tapi, karena set adalah tipe data tidak berurut (unordered), maka kita tidak bisa menggunakan indeks. Set tidak mendukung indeks ataupun slicing. Untuk menambah satu anggota ke dalam set, kita bisa menggunakan fungsi `add()`, dan untuk menambahkan beberapa anggota sekaligus kita bisa menggunakan fungsi `update()`. List, tuple, maupun string bisa digunakan sebagai masukan dari fungsi `update()`. Contoh program Mengubah Anggota Set :



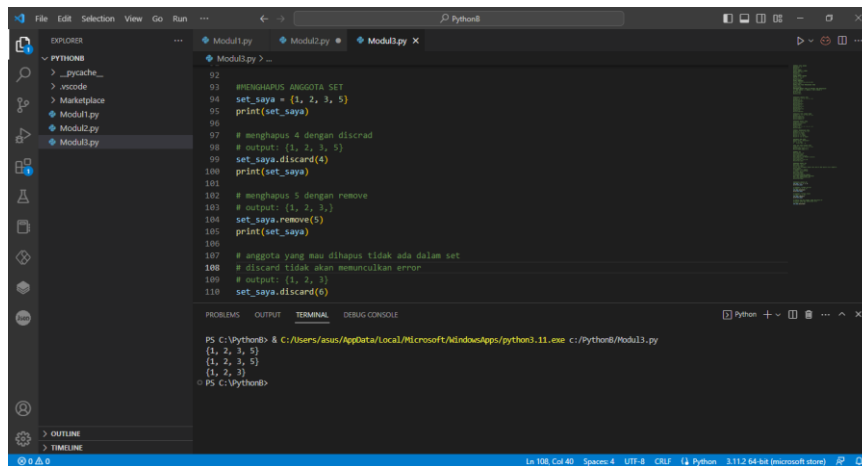
```
78 #MENUBAH ANGGOTA SET
79 set_saya = {1,2,3}
80 print(set_saya)
81 # jika kita hilangkan tanda # dari baris 9 akan muncul error TypeError
82 #set_saya[0]
83 # menambah satu anggota
84 # input: {1,2,3,4,5,6}
85 set_saya.add(4)
86 print(set_saya)
87 # menambah beberapa anggota
88 #set akan menghilangkan duplikasi
89 set_saya.update([3,4,5,6])
90 print(set_saya)
91
```

Terminal Output:

```
PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python\Modul3.py
{1, 2, 3}
{1, 2, 3, 4}
{1, 2, 3, 4, 5, 6}
```

3. Menghapus Anggota Set

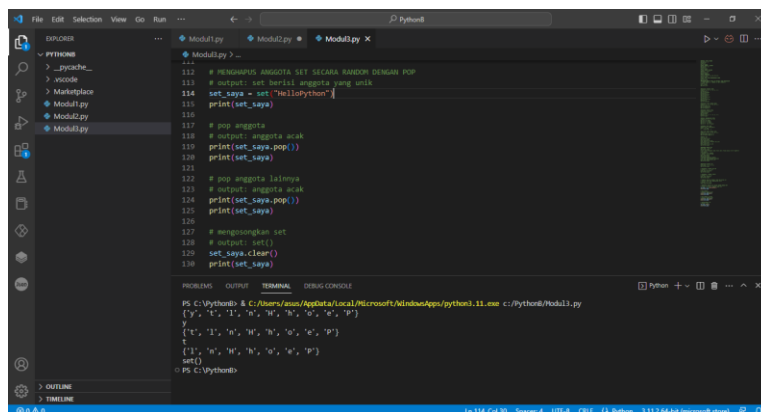
Kita bisa menghapus anggota set dengan menggunakan fungsi `discard()` dan `remove()`. Contoh program Menghapus Anggota Set



```
92
93 #MENGGHAPUS ANGGOTA SET
94 set_saya = {1, 2, 3, 5}
95 print(set_saya)
96
97 # menghapus 4 dengan discard
98 # output: {1, 2, 3, 5}
99 set_saya.discard(4)
100 print(set_saya)
101
102 # menghapus 5 dengan remove
103 # output: {1, 2, 3,}
104 set_saya.remove(5)
105 print(set_saya)
106
107 # anggota yang mau dihapus tidak ada dalam set
108 # discard tidak akan memunculkan error
109 # output: {1, 2, 3}
110 set_saya.discard(6)
```

PS C:\Python8> C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python8\Modul3.py
{1, 2, 3, 5}
{1, 2, 3, 5}
{1, 2, 3}
PS C:\Python8>

Menghapus Anggota Set Secara Random dengan pop(). Contoh program :



```
112
113 # MENGHAPUS ANGGOTA SET SECARA RANDOM DENGAN POP
114 set_saya = set("HelloPython")
115 print(set_saya)
116
117 # pop anggota
118 # output: anggota acak
119 print(set_saya.pop())
120 print(set_saya)
121
122 # pop anggota lainnya
123 # output: anggota acak
124 print(set_saya.pop())
125 print(set_saya)
126
127 # Mengosongkan set
128 # output: set()
129 set_saya.clear()
130 print(set_saya)
```

PS C:\Python8> C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:\Python8\Modul3.py
{'y', 't', 'l', 'o', 'H', 'e', 'n', 'P', 'o', 'P'}
{'t', 'l', 'o', 'H', 'e', 'n', 'P', 'o'}
{'t', 'l', 'o', 'H', 'e', 'n', 'P'}
set()
PS C:\Python8>

4. Operasi Gabungan (Union)

Operasi (Union) merupakan operasi gabungan dari suatu himpunan contoh himpunan A dan B. Gabungan dapat dibuat dengan menggunakan operator palang (|). Selain itu juga bisa dilakukan dengan menggunakan fungsi union(). Contoh program Operasi Gabungan (Union) dengan Set

```

131
132 # membuat union set A and B
133 A = {1, 2, 3, 4, 5}
134 B = {4, 5, 6, 7, 8}
135
136 # Gabungkan menggunakan operator
137 # output: {1, 2, 3, 4, 5, 6, 7, 8}
138 print(A | B)
139
140 # Menggunakan fungsi union()
141 # output: {1, 2, 3, 4, 5, 6, 7, 8}
142 A.union(B)
143
144 # output: {1, 2, 3, 4, 5, 6, 7, 8}
145 B.union(A)

```

PROBLEMS OUTPUT

Open file in editor (ctrl + click)

PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python11.exe c:/Python/Modul1.py
{1, 2, 3, 4, 5, 6, 7, 8}
C:\Python>

5. Operasi Irisan (Intersection)

Operasi Irisan dilakukan dengan menggunakan operator jangkar (&). Irisan juga bisa dilakukan dengan menggunakan fungsi intersection(). Contoh program Operasi Irisan (Intersection) dengan Set :

```

146
147 # membuat irisan set A dan B
148 A = {1, 2, 3, 4, 5}
149 B = {4, 5, 6, 7, 8}
150
151 # Irisan menggunakan operator &
152 # output: {4, 5}
153 print(A & B)
154 # Menggunakan fungsi intersection()
155 # output: {4, 5}
156 A.intersection(B)
157
158 # Output: {4, 5}
159 B.intersection(A)

```

PROBLEMS OUTPUT

PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python11.exe c:/Python/Modul1.py
{4, 5}
C:\Python>

6. Operasi Selisih (Difference)

Operasi Selisih dilakukan dengan menggunakan operator kurang (-). Bisa juga dengan menggunakan fungsi difference(). Contoh program Operasi Selisih (Difference) dengan Set:

```

160
161 # membuat selisih (difference) A dan B
162 A = {1, 2, 3, 4, 5}
163 B = {4, 5, 6, 7, 8}
164
165 # Menggunakan operator - pada A
166 # Output: {1, 2, 3}
167 print(A - B)
168
169 # Output: {1, 2, 3}
170 A.difference(B)
171
172 # menggunakan operator - B
173 # Output: {6, 7, 8}
174 print(B - A)
175
176 # Output: {6, 7, 8}
177 B.difference(A)

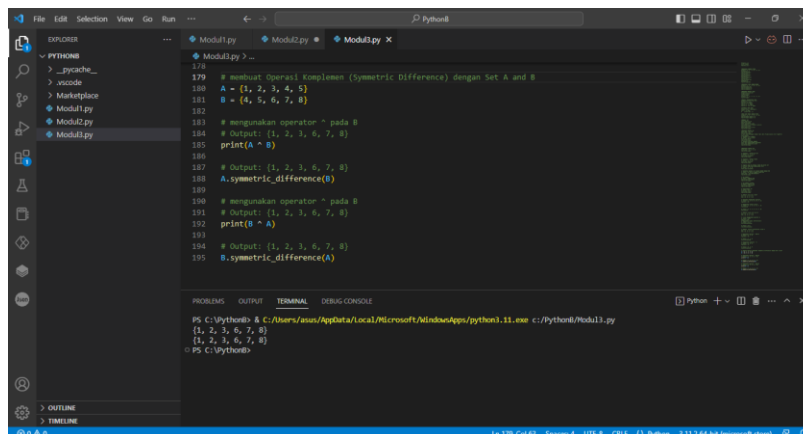
```

PROBLEMS OUTPUT

PS C:\Python> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python11.exe c:/Python/Modul1.py
{1, 2, 3}
{6, 7, 8}
C:\Python>

7. Operasi Komplemen (Symmetric Difference)

Operasi komplement dilakukan dengan menggunakan operator \wedge . Bisa juga dengan menggunakan fungsi `symmetric_difference()`. Contoh program Operasi Komplement (Symmetric Difference) dengan Set



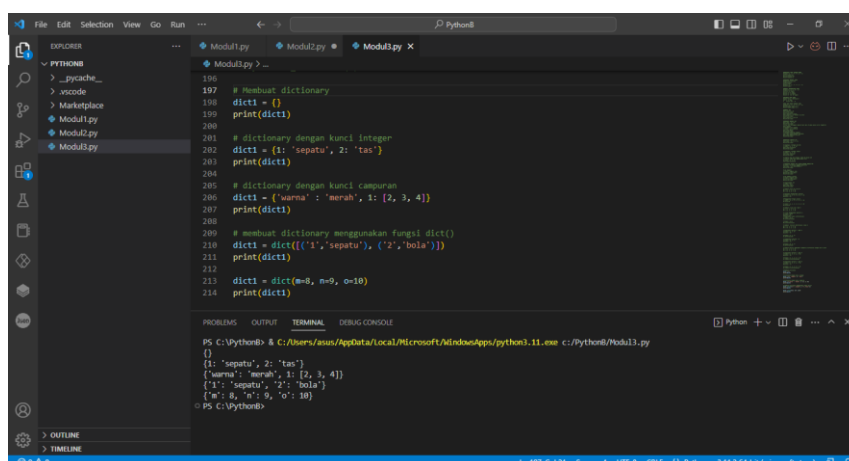
```
178
179 # membuat Operasi Komplement (Symmetric Difference) dengan Set A and B
180 A = {1, 2, 3, 4, 5}
181 B = {4, 5, 6, 7, 8}
182
183 # menggunakan operator ^ pada B
184 # Output: {1, 2, 3, 6, 7, 8}
185 print(A ^ B)
186
187 # Output: {1, 2, 3, 6, 7, 8}
188 A.symmetric_difference(B)
189
190 # menggunakan operator ^ pada A
191 # Output: {1, 2, 3, 6, 7, 8}
192 print(B ^ A)
193
194 # Output: {1, 2, 3, 6, 7, 8}
195 B.symmetric_difference(A)
```

Dictionary

Tipe data yang anggotanya terdiri dari pasangan kunci:nilai (key:value). Dictionary bersifat tidak berurut (unordered) sehingga anggotanya tidak memiliki indeks.

1. Membuat Dictionary

Dictionary dibuat dengan menempatkan anggotanya di dalam tanda kurung kurawal {}, dipisahkan oleh tanda koma. Anggota dictionary terdiri dari pasangan kunci:nilai. Kunci harus bersifat unik, tidak boleh ada dua kunci yang sama dalam dictionary. Contoh program Membuat Dictionary :



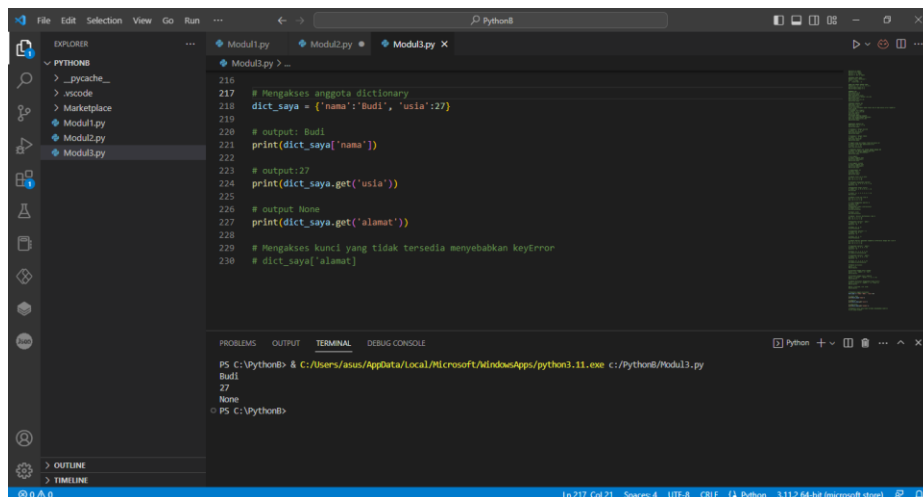
```
196
197 # membuat dictionary
198 dict1 = {}
199 print(dict1)
200
201 # dictionary dengan kunci integer
202 dict1 = {1: 'sepatu', 2: 'tas'}
203 print(dict1)
204
205 # dictionary dengan kunci campuran
206 dict1 = {'warna': 'merah', 1: [2, 3, 4]}
207 print(dict1)
208
209 # membuat dictionary menggunakan fungsi dict()
210 dict1 = dict([('1', 'sepatu'), ('2', 'bola')])
211 print(dict1)
212
213 dict1 = dict(m=8, n=9, o=10)
214 print(dict1)
```

2. Mengakses Anggota Dictionary

Dictionary tidak menggunakan indeks. Anggota dictionary diakses dengan menggunakan kuncinya.bisa juga diakses dengan menggunakan fungsi `get()`.

Dengan menggunakan fungsi `get()`, bila kunci tidak ada di dalam dictionary, maka akan dikembalikan `None`. Bila tidak menggunakan fungsi `get()`, maka akan terjadi error `KeyError` bila kunci yang ingin diakses tidak ada di dalam dictionary.

Contoh program Mengakses Anggota



```
216
217 # Mengakses anggota dictionary
218 dict_saya = {'nama': 'Budi', 'usia': 27}
219
220 # output: Budi
221 print(dict_saya['nama'])
222
223 # output: 27
224 print(dict_saya.get('usia'))
225
226 # output: None
227 print(dict_saya.get('alamat'))
228
229 # Mengakses kunci yang tidak tersedia menyebabkan KeyError
230 dict_saya['alamat']
```

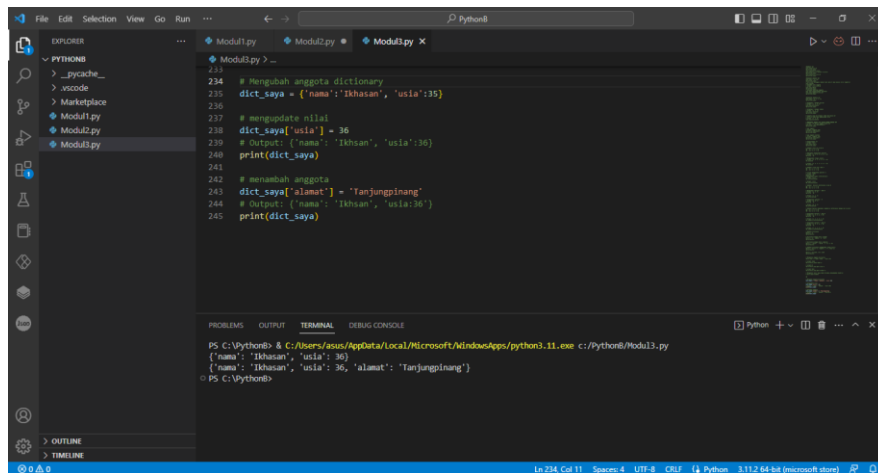
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Python8> & C:\Users\asus\AppData\Local\Microsoft\WindowsApps\python3.11.exe c:/Python8/Modul3.py
Budi
27
None
PS C:\Python8>
```

Ln 217, Col 21 Spaces: 4 UTF-8 CRUF Python 3.11.2 64-bit (microsoft store)

3. Mengubah Anggota Dictionary

Dictionary bersifat mutable. Kita bisa menambahkan atau mengubah nilai dari anggotanya menggunakan operator penugasan. Contoh program Mengubah Anggota Dictionary

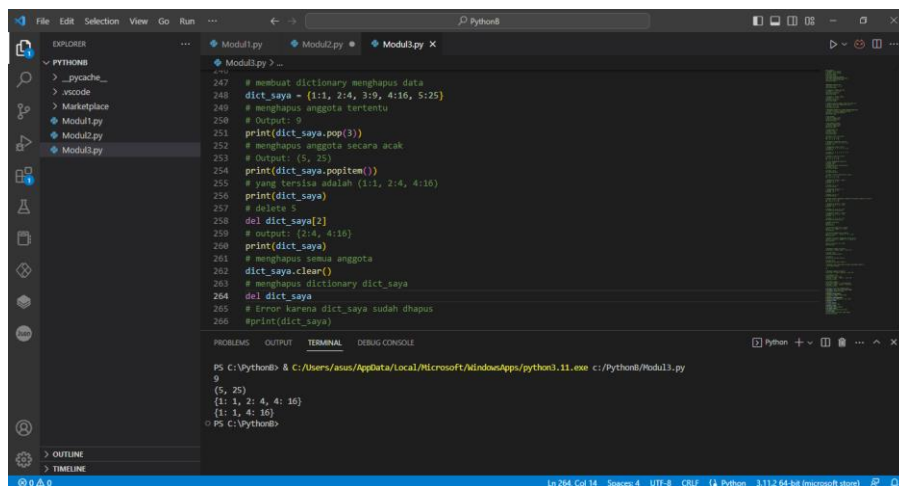


```
234 # Mengubah anggota dictionary
235 dict_saya = {'nama': 'Ikhsan', 'usia': 35}
236
237 # mengupdate nilai
238 dict_saya['usia'] = 36
239 # Output: {'nama': 'Ikhsan', 'usia': 36}
240 print(dict_saya)
241
242 # menambah anggota
243 dict_saya['alamat'] = 'Tanjungpinang'
244 # Output: {'nama': 'Ikhsan', 'usia': 36, 'alamat': 'Tanjungpinang'}
245 print(dict_saya)
```

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul3.py
{'nama': 'Ikhsan', 'usia': 36}
{'nama': 'Ikhsan', 'usia': 36, 'alamat': 'Tanjungpinang'}
```

4. Menghapus Anggota Dictionary

Kita bisa menghapus anggota tertentu pada dictionary dengan menggunakan fungsi `pop()`. Fungsi ini menghapus anggota dengan mengembalikan kunci dari anggota tersebut. Fungsi lain, `popitem()` digunakan untuk menghapus anggota acak dari dictionary. Untuk menghapus semua anggota dictionary, bisa menggunakan fungsi `clear()`. Selain itu kita juga bisa menggunakan kata kunci `del` untuk menghapus anggota tertentu atau menghapus dictionary itu sendiri. Contoh program Menghapus Anggota Dictionary



```
247 # membuat dictionary menghapus data
248 dict_saya = {'1': 1, '2': 4, '3': 9, '4': 16, '5': 25}
249 # menghapus anggota tertentu
250 # Output: 9
251 print(dict_saya.pop('3'))
252 # menghapus anggota secara acak
253 # Output: (5, 25)
254 print(dict_saya.popitem())
255 # yang tersisa adalah {'1': 1, '2': 4, '4': 16}
256 print(dict_saya)
257 # delete 5
258 del dict_saya[2]
259 # output: {'1': 1, '4': 16}
260 print(dict_saya)
261 # menghapus semua anggota
262 dict_saya.clear()
263 # menghapus dictionary dict_saya
264 del dict_saya
265 # Error karena dict_saya sudah dhapus
266 #print(dict_saya)
```

```
PS C:\Python8> & C:/Users/asus/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Python8/Modul3.py
9
{'1': 1, '2': 4, '4': 16}
{'1': 1, '4': 16}
Traceback (most recent call last):
  File "c:/Python8/Modul3.py", line 266, in <module>
    #print(dict_saya)
    ~~~~~~^
NameError: name 'dict_saya' is not defined
```