

### Soal Pretest

1. Perhatikan potongan kode berikut!

```
int main()
{
int i = 3;
while(i < 3) {
printf("W");
}
return 0;
}
```

Output dari kode program di atas adalah ..

- A. WWWWWW
- B. no output
- C. WW
- D. W
- E. WWW

### Jawab B

**Penjelasan:** sebelum masuk ke **while**, nilai **i** sudah diinisialisasi dengan 3, jadi isi dari **i<3** bernilai **FALSE**, sehingga perintah di dalam while tidak pernah dieksekusi.

---

2. Perhatikan potongan kode berikut!

```
int main()
{
int i, j;
for (i = 0, j = 4 - 1; j > 0; j--, i++) {
printf("#%d", i);
}
return 0;
}
```

Output dari kode program di atas adalah ...

- A. #0#1#2#3
- B. #4#3#2#1
- C. #0#1#2#3#4
- D. #0#1#2
- E. #4#3#2

**Jawab: D**

**Penjelasan:** perhatikan pada perintah perulangan menggunakan **for**. Variabel yang menjadi pencacah pada perulangan tersebut adalah **variable j**, dimana perulangannya dimulai dari **j=3** sampai **j>0**. Jadi perintah di dalam perulangan ini dieksekusi sebanyak 3 kali yaitu ketika **j=3,2,dan 1**. Sedangkan perintah yang dilakukan adalah mencetak isi dari **variabel i**, dimana i dimulai dari angka 0, dan terus bertambah. Karena perulangan dilakukan sebanyak 3 kali, i dicetak sebanyak 3 kali yaitu ketika **i= 0, 1, dan 2**.

---

3. Perhatikan potongan kode berikut!

```
int main()
{
int i = 0;
while(i++ < 3) {
if (i == 2)
break;
printf("W");
}
return 0;
}
```

Output dari kode program di atas adalah ...

- A. WWWWWW
- B. no output
- C. WW
- D. W
- E. WWW

Jawab: D

Penjelasan: awalnya i diinisialisasi dengan 0, ketika masuk ke **while**, **variabel i** diincrement dengan 1 akhirnya i=1, karena **1<3** bernilai **TRUE**, maka perintah di dalam while dieksekusi. Lalu masuk ke if. Karena **i==2** bernilai **FALSE**, maka perintah di bawahnya if tidak dieksekusi, lanjut ke perintah berikutnya yaitu **printf ("W")**. Dari sini hasilnya W dicetak satu kali. Kemudian kembali lagi ke while, i=2. Karena **2<3** bernilai **TRUE**, maka perintah di dalam while dieksekusi. Lalu masuk ke if. Karena **i==2** bernilai **TRUE**, maka perintah di bawahnya if yaitu **break**, dieksekusi. Karena perintah **break** yang dieksekusi, maka program langsung keluar dari perulangan secara paksa. Jadi W hanya dicetak sebanyak satu kali.

---

4. Perhatikan potongan kode berikut!

```
int main()
{
```

```
int i = 10, j = 3;
if ( i < 5 || j > 0)
{
j++;
i++;
}
printf("%d %d", i, j);
return 0;
}
```

Output dari kode program di atas adalah ...

- A. 11 3
- B. 10 3
- C. 10 5
- D. 11 5
- E. 11 4

**Jawab: E**

**Penjelasan:** awalnya i=10, dan j=3, kondisi **10<5 OR 3>0** bernilai TRUE, sehingga perintah di dalam IF dieksekusi. Jadinya j++, menghasilkan j=4, dan i++ menghasilkan i=11. Jadi ketika i dan j dicetak hasilnya adalah 11 4

---

5. Perhatikan potongan kode berikut!

```
int main()
{
int i = 0;
while(++i < 3) {
printf("W");
}
return 0;
}
```

Output dari kode program di atas adalah ...

- A. WWWWWW
- B. no output
- C. WW
- D. W
- E. WWW

Jawab: C

Penjelasan: awalnya *i* diinisialisasi dengan 0, ketika masuk ke **while**, variabel *i* diincrement dengan 1 akhirnya *i*=1, karena  $1 < 3$  bernilai **TRUE**, maka perintah di dalam while dieksekusi. Lalu masuk ke perintah berikutnya yaitu **printf ("W")**. Dari sini hasilnya W dicetak satu kali. Kemudian kembali lagi ke while, *i*=2. Karena  $2 < 3$  bernilai **TRUE**, maka perintah di dalam while dieksekusi. Lalu masuk ke perintah berikutnya yaitu **printf ("W")**. Dari sini hasilnya W dicetak satu kali lagi. Kemudian kembali lagi ke while, *i*=3. Karena  $3 < 3$  bernilai **FALSE**, maka perintah di dalamnya sudah tidak dieksekusi. Jadi totalnya W di cetak sebanyak 2 kali.

---

6. Perhatikan potongan kode berikut!

```
int main()
{
int i = 3;
do {
printf("P");
} while(i > 3);
return 0;
}
```

Output dari kode program di atas adalah ...

- A. P P P P P P
- B. P
- C. no output
- D. P P P
- E. P P

Jawab: B

Penjelasan: awalnya *i* diinisialisasi dengan 3. Karena menggunakan perintah **do**, maka perintah di dalam do yaitu **printf("P")** langsung dieksekusi. Kemudian masuk ke **while**. Kondisi  $3 > 3$  bernilai **FALSE**, sehingga perintah di dalam **do** tidak dieksekusi lagi. Jadi totalnya P dicetak sebanyak satu kali.

---

7. Perhatikan potongan kode berikut!

```
int main()
{
int val = 4;
if (val != 10) {
printf("*");
} else if (val == 4) {
printf("**");
}
```

```
} else if (val == -4) {  
printf("****");  
} else {  
printf("*****");  
}  
printf("*");  
return 0;  
}
```

Output dari kode program di atas adalah ..

- A. \*
- B. \*\*\*\*\*
- C. \*\*
- D. \*\*\*
- E. \*\*\*\*

Jawab: C

---

Penjelasan: awalnya variabel **val=4**. Lalu masuk ke **if** yang pertama. Kondisi **4!=10** bernilai **TRUE**, sehingga perintah **printf("\*")** dieksekusi menghasilkan 1 bintang. Karena berikutnya berupa **else**, jadi sudah tidak dieksekusi. Sampai yang terakhir keluar dari if ketemu perintah **printf("\*")** dieksekusi lagi. Jadi totalnya tanda (\*) dicetak sebanyak 2 kali.

---

8. Perhatikan potongan kode berikut!

```
int main()  
{  
char kar = 'A';  
if (kar == 'a') {  
printf("It's an A");  
}  
else {  
printf("It's not an A");  
}  
}
```

Output dari kode program di atas adalah ..

- A. It's not an Alt's an A
- B. It's an A
- C. It's an Alt's an A
- D. It's not an A
- E. It's an Alt's not an A

Jawab: D

---

Penjelasan: di dalam c/c++, huruf 'A' dan 'a' dianggap berbeda. Pada program tersebut variabel **kar='A'**. Ketika masuk **if** yang pertama, kondisi '**A'=='a'**' akan bernilai **FALSE**, sehingga program langsung mengeksekusi perintah di dalam else yaitu perintah **printf("It's not an A")**. Jadi keluarnya adalah pilihan D.

---

9. Perhatikan potongan kode berikut!

```
int main()
{
for (int i = 0; i < 3; i++) {
printf("$");
}
}
```

Output dari kode program di atas adalah ...

- A. no output
- B. \$\$
- C. \$\$\$
- D. \$\$\$\$
- E. \$

Jawab: C

Penjelasan: perhatikan pada perintah perulangan menggunakan **for**. Variabel yang menjadi pencacah pada perulangan tersebut adalah **variable i**, dimana perulangannya dimulai dari **i=0** sampai **i<3**. Jadi perintah di dalam perulangan ini dieksekusi sebanyak 3 kali yaitu ketika **i=0,1,dan 2**. Sedangkan perintah yang dilakukan adalah **printf("\$")**. Karena perulangan dilakukan sebanyak 3 kali, maka tanda (\$) dicetak sebanyak 3 kali.

---

10. Perhatikan potongan kode berikut!

```
int main()
{
int i = 1;
if (i < 10)
```

```
printf("%d", i);  
else  
printf("%d", i);  
return 0;  
}
```

Output dari kode program di atas adalah ..

- A. 1
- B. 11111111111
- C. 1111111111
- D. 11
- E. 111111111

Jawab: A

Penjelasan: awalnya variabel  $i=1$ . Lalu masuk ke **if** yang pertama. Kondisi  **$1 < 10$**  bernilai **TRUE**, sehingga perintah **printf("%d", i)** yaitu perintah untuk mencetak nilai  $i$  dieksekusi. Karena perikutnya berupa **else**, jadi sudah tidak dieksekusi.

---

11. Perhatikan potongan kode berikut!

```
void calc(int val)  
{  
val += 10;  
printf("%d", val);  
}  
int main()  
{  
int num = 3;  
calc(num);  
printf("%d", num);  
}
```

Output dari kode program di atas adalah ..

- A. 1313
- B. 33
- C. 1010
- D. 103
- E. 133

Jawab: E

Penjelasan: awalnya variabel **num=3**. Lalu memanggil fungsi `calc`, dengan parameter input variabel `num`. Di fungsi `calc`, nilai `val=3`, kemudian nilai `val` ditambah dengan 10, menjadi 13, kemudian nilai `val` di cetak. Kemudian kembali lagi ke fungsi `main` mencetak nilai `num`. Jadi outputnya adalah 133

---

12. Perhatikan potongan kode berikut!

```
int main()
{
int i = 0;
do {
printf("P");
} while(i++ < 4);
return 0;
}
```

Output dari kode program di atas adalah ..

- A. P
- B. P P P P P
- C. P P P P
- D. P P P
- E. no output

jawab: C

Penjelasan: awalnya `i` diinisialisasi dengan 0. Karena menggunakan perintah **do**, maka perintah di dalam `do` yaitu **printf("P")** langsung dieksekusi. Kemudian masuk ke **while dengan i diincrement**. Kondisi **1<4** bernilai **TRUE**, sehingga perintah di dalam **do** dieksekusi lagi. Kemudian masuk ke **while dengan i diincrement**. Kondisi **2<4** bernilai **TRUE**, sehingga perintah di dalam **do** dieksekusi lagi. Kemudian masuk ke **while dengan i diincrement**. Kondisi **3<4** bernilai **TRUE**, sehingga perintah di dalam **do** dieksekusi. Kemudian masuk ke **while dengan i diincrement**. Kondisi **4<4** bernilai **FALSE**, sehingga perintah di dalam **do TIDAK DIEKSEKUSI**. Jadi totalnya P dicetak sebanyak 4 kali.

---

13. Perhatikan potongan kode berikut!

```
int main()
{
int i = 1;
if (i < 10)
printf("%d", i);
return 0;
```



```
}
```

Output dari kode program di atas adalah ...

- A. 1111111111
- B. 1
- C. 1111111111
- D. 11
- E. 1111111111

Jawab: B

Penjelasan: awalnya variabel **i=1**. Lalu masuk ke **if** yang pertama. Kondisi **1<10** bernilai **TRUE**, sehingga perintah **printf("%d", i)** yaitu perintah untuk mencetak nilai **i** dieksekusi.