**Analisis Kasus 4**

1. Kesalahan yang pertama adalah x, y, dan jumlah menggunakan tipe data integer yang seharusnya menggunakan float
2. Kesalahan yang kedua, menggunakan pointer, sedangkan yang diminta outputnya adalah jumlah bertipe data float

**Notasi Algoritma :**

**Program Coba\_Pointer {program latihan tentang pointer}**

**Kamus local :**

x,y,jumlah : float

\*ptrA: float

\*ptrB: float

**Algoritma :**

x ← 10

y ← 20

\*ptrA ← 20.5

\*ptrB ← 30.5

jumlah ← x + y

output(jumlah)

**Analisis Kasus 5**

Program nomor 5 merupakan modifikasi dari array yang ditampung dalam pointer, ditambahnya variable pointer tidak menambah memori pada suatu program karena sudah memiliki value dari array yang ditampung oleh si pointer. Program ini memiliki maximal input array sebanyak 5 kali, lalu setelah itu array akan ditampung ke pointer. Array index kedua akan diubah ke value yang baru, maka dideklarasikan untuk ditampung ke pointer ptr, selanjutnya ptr mengambil address array index kedua. \*ptr menunjukkan bahwa pointer sudah mulai menampung value baru. Terakhir Output dikeluarkan dari array data i atau index i

**Notasi Algoritma :**

**Program Coba\_Pointer {program modifikasi array }**

**Kamus local :**

Max = 5 : integer

Arr : integer

\*ptr : integer

**Algoritma :**

**For int I = 0, I < max, I++ do**

Input(arr)

Arr[2] ← 100

Int\* ptr ← &arr[2]

\*ptr ← 100

**For int I = 0, I < max, I++ do**

output(arr[i])

return 0

Analisis Kasus 6

1. ++ptr

Kesimpulan : Hanya dua elemen terakhir yang di loop 2 kali

1. \*(ptr++ )

Kesimpulan : Akan ada variabel yang dikhianati atau di tinggal karena efek samping dari postfix

1. \*(++ptr )

Kesimpulan : Hanya dua element terakhir yang di tampilkan tanpa looping