

```
public static int compute(String word) {  
    char[] letters = {'a', 'e', 'i', 'o', 'u'};  
    int result = 0;  
    for (int i = 0; i < word.length(); i++) {  
        for (int j = 0; j < letters.length; j++) {  
            if (word.charAt(i) == letters[j]) {  
                result++;  
            }  
        }  
    }  
    return result;  
}
```

The diagram illustrates the control flow of the provided Java code. Nodes, represented by colored circles, are placed at various points in the code: at the start of the function, before the letters array declaration, before the result variable declaration, at the start of the outer loop, at the start of the inner loop, before the if-statement, after the if-statement, after the inner loop, after the outer loop, before the return statement, and at the end of the function. Edges, shown as lines, connect these nodes to represent the flow of execution: from the start to the array declaration, then to the result declaration, then to the outer loop. From the outer loop, the flow branches into the inner loop. The inner loop branches into the if-statement. The if-statement branches into a path that increments the result and a path that skips the increment. Both paths merge and flow back to the start of the inner loop. After the inner loop, the flow goes to the end of the outer loop, then to the return statement, and finally to the end of the function.