

Analisis Kesulitan Mahasiswa Prodi PTIK FKIP UNS dalam Penyelesaian Masalah dengan Pemrograman

Dwi Maryono

Program Studi Pendidikan TIK FKIP UNS
Kampus FKIP Pabelan Jl. Ahmad Yani No.200 Pabelan Sukoharjo
email:dwimarus@gmail.com

Abstrak

Penelitian ini bertujuan untuk mengetahui kesulitan siswa dalam menyelesaikan permasalahan dengan Pemrograman. Langkah-langkah penyelesaian masalah menurut Polya adalah memahami permasalahan (*understand the problem*), menyusun rencana penyelesaian (*devising a plan*), melaksanakan rencana penyelesaian (*carrying out the plan*), dan memeriksa kembali penyelesaian (*looking back*).

Penelitian ini merupakan penelitian kualitatif dengan populasi adalah mahasiswa semester 3 Prodi PTIK FKIP. Teknik pengambilan sampel adalah *purposive sampling*, dengan pendekatan *snowball*, di mana sampel diambil satu persatu untuk mendapatkan data mengenai kesulitan mahasiswa dalam menerapkan langkah penyelesaian masalah dengan Pemrograman. Pengambilan sampel berhenti jika data yang diperoleh sudah jenuh.

Hasil penelitian menunjukkan bahwa kesulitan yang dialami mahasiswa adalah sebagai berikut. Pada tahap memahami permasalahan, terdapat mahasiswa yang tidak mampu menterjemahkan masalah yang diberikan dengan bahasa pemrograman, seperti menurunkan variabel dan tipe data yang dibutuhkan. Dalam proses design, banyak mahasiswa yang tidak menuliskan rancangan program baik dalam flowchart, pseudocode maupun bahasa natural. Banyak di antara mereka yang langsung menuliskan kode program tanpa merancang algoritmanya terlebih dahulu. Hal ini disebabkan karena dalam soal tidak diperintahkan untuk membuat algoritma terlebih dahulu dan beberapa siswa tidak terbiasa menuliskan algoritma sebelum coding. Dalam tahap writing, beberapa masih kurang menguasai sintaks sehingga terhambat dalam proses penyelesaian masalah. Dalam tahap review kesulitan yang dihadapi adalah kurangnya penguasaan debugging, baik untuk membenarkan kesalahan penulisan maupun kesalahan logika atau algoritma.

Abstrak: Analisis kesulitan, *problem solving*, pemrograman

1. PENDAHULUAN

Pemrograman dasar merupakan mata kuliah yang penting dalam mempelajari ilmu komputer terutama yang terkait dengan perancangan perangkat lunak atau aplikasi. Mata kuliah ini memberikan dasar-dasar algoritma dan pemrograman yang sangat berguna bagi mahasiswa ketika mempelajari mata kuliah lain yang terkait dengan pemrograman seperti pemrograman web, pemrograman berorientasi obyek, *mobile programming* dan sebagainya. Ada beberapa bahasa yang dapat digunakan dalam membelajarkan pemrograman dasar, namun di antaranya yang sering digunakan adalah bahasa pemrograman Pascal. Bahasa pemrograman ini dipilih karena relatif lebih mudah dipelajari karena strukturnya yang sederhana dengan menggunakan sintaks sebagaimana bahasa manusia, dengan meminimumkan simbol-simbol. Oleh karena itu Pascal mudah dipelajari

(https://www.tutorialspoint.com/pascal/pascal_tutorial.pdf).

Bagi mahasiswa, meskipun merupakan mata kuliah dasar, pemrograman dasar dianggap merupakan salah satu mata kuliah yang sulit, karena menggabungkan beberapa kemampuan, seperti logika, algoritma dan kemampuan pemecahan masalah. Mahasiswa diharapkan tidak sekedar mampu menuliskan program tapi juga diharapkan mampu menyelesaikan permasalahan dengan menggunakan pemrograman. Menurut Becker dan Shimada dalam Mcintosh (2000), masalah mempunyai dua ciri, yaitu menantang pikiran (*challenging*) dan non routine, tidak secara otomatis diketahui penyelesaiannya. Menurut Polya, dalam Voskoglou (2012), menyelesaikan masalah adalah mencari jalan keluar dari suatu kesulitan, melalui berbagai rintangan untuk mencapai tujuan yang tidak dapat dipahami secara cepat.

Terkadang dalam menyelesaikan permasalahan, mahasiswa terpaku pada masalah

routine, yang pernah diberikan sebelumnya. Ketika mereka menjumpai soal atau permasalahan yang berbeda atau dengan sedikit modifikasi, mereka kesulitan mengerjakan. Ini menunjukkan bahwa kemampuan penyelesaian masalah masih kurang. Hal ini tentu akan menjadi masalah, karena di dunia nyata seorang *programmer* dihadapkan pada berbagai macam permasalahan, di mana diantaranya banyak yang merupakan permasalahan *non routine*. Kemampuan pemecahan masalah yang baik menjadi bekal yang kuat bagi mahasiswa dalam menghadapi dunia kerja.

Secara umum, langkah-langkah penyelesaian masalah menurut Polya (1945), meliputi memahami permasalahan (*understand the problem*), menyusun rencana penyelesaian (*devising a plan*), implementasi rencana penyelesaian (*carrying out the plan*), dan memeriksa kembali penyelesaian (*looking back*). Meskipun Polya menerapkan langkah-langkah tersebut pada masalah matematika, hal ini dapat diberlakukan juga pada penyelesaian masalah dalam pemrograman (Barnes, dkk, 1997). Dalam penyelesaian masalah pemrograman, tahap pertama adalah memahami permasalahan. Dalam tahap ini seorang *programmer* harus mampu memahami permasalahan secara keseluruhan, mengidentifikasi tiap-tiap deskripsi masalah yang diberikan. Pada tahap ini programmer harus mampu mengidentifikasi spesifikasi program yang harus dibuat, input dan output dari masalah yang diberikan. Hasil pemahaman masalah ini digunakan untuk menentukan strategi apa yang akan diterapkan. Langkah kedua, membuat rancangan penyelesaian. Pada tahap ini programmer membuat langkah-langkah penyelesaian sebelum membuat program. Rancangan penyelesaian dapat disusun berdasarkan masalah lain yang terkait atau mirip dengan permasalahan yang diberikan. Langkah-langkah penyelesaian dapat dituliskan dalam

bentuk algoritma, baik dengan bahasa natural, *pseudocode* maupun dengan *flowchart*. Langkah ketiga adalah tahap implementasi. Pada tahap ini programmer menuliskan kode program berdasarkan langkah-langkah yang sudah dibuat. Pada tahap ini penguasaan sintaks bahasa pemrograman sangat berpengaruh. Semakin baik penguasaan sintaks maka dapat meminimalkan error, sehingga program dapat cepat diselesaikan. Langkah terakhir adalah memeriksa kembali. Pada tahap ini programmer memeriksa apakah program yang sudah dibuat sudah memberikan penyelesaian yang tepat atau masih ada kesalahan. Kesalahan yang sering terjadi adalah kesalahan sintaks, terkait dengan kesalahan penulisan kode program dan kesalahan algoritmik yang terkait dengan langkah-langkah penyelesaian program.

Dari hasil pengamatan yang dilakukan terhadap mahasiswa PTIK, kemampuan penyelesaian masalah mahasiswa khususnya PTIK masih belum begitu baik. Hal ini terlihat pada mata kuliah Algoritma dan Struktur Data, ketika dihadapkan pada masalah baru mahasiswa sering kebingungan untuk menentukan penyelesaiannya. Hal ini tentunya akan berpengaruh terhadap mata kuliah lain terutama yang terkait dengan pemrograman. Oleh karena itu penelitian ini fokus pada kemampuan penyelesaian masalah pemrograman pada mahasiswa PTIK dan kesulitan-kesulitan yang dihadapi, sehingga dapat dijadikan acuan dalam menyusun strategi pembelajaran yang tepat.

2. METODE

Penelitian ini merupakan penelitian kualitatif dengan populasi adalah mahasiswa semester 3 Prodi PTIK FKIP. Teknik pengambilan sampel adalah *purposive sampling*, dengan pendekatan *snowball*, di mana sampel diambil satu persatu untuk mendapatkan data mengenai

kesulitan mahasiswa dalam menerapkan langkah penyelesaian masalah dengan Pemrograman. Pengambilan sampel berhenti jika data yang diperoleh sudah jenuh.

3. HASIL DAN PEMBAHASAN

Dalam penelitian digunakan sebuah soal untuk mengukur tingkat kemampuan penyelesaian masalah dan kesulitan-kesulitan yang dialami oleh mahasiswa dalam penyelesaian masalah

Permasalahan:

Pak Rebo mempunyai dua barisan kandang kambing yang bernomor $A_1 - A_n$ dan $B_1 - B_n$ yang saling berhadapan. Tiap-tiap kandang berisi seekor kambing yang sudah di dihitung berat badannya. Berat dari tiap kambing dikalungkan pada lehernya. Menjelang Idul Adha tiba, pak Rebo ingin menjual separuh dari kambing-kambingnya dengan cara memilih kambing tergemuk dari tiap dua kandang yang saling berhadapan. Seluruh kambing yang telah dipilih kemudian dimasukkan ke kandang $C_1 - C_n$ sesuai dengan urutan pengambilan dari kandang $A_1 - A_n$ atau $B_1 - B_n$. Buatlah program untuk membantu Pak Rebo memilih kambing-kambing yang akan dijual dengan menuliskan asal barisan kandang dan bobot kambing yang dipilih.

Input dari program adalah banyak kandang dalam satu baris, berat kambing-kambing dalam barisan kandang A dan berat kambing-kambing dalam barisan kandang B. Outputnya adalah kode barisan asal kambing-kambing yang dipilih dan total berat.

Contoh:

Banyak kandang adalah : 8

Kandang A

15 22 18 19 24 15 18 20

Kandang B

11 25 16 18 27 16 14 17

Output

A15 B25 A18 A19 B27 B16 A18 A20

Total Berat kambing: 158

Soal tersebut diberikan kepada mahasiswa PTIK semester 3 dengan asumsi yang sudah menempuh baik mata kuliah pemrograman dasar dan struktur algoritma, dengan bahasa Pascal. Hasil pekerjaan kemudian dianalisis untuk kemudian dipilih berapa sampel untuk menggali kemampuan penyelesaian masalah dan permasalahan yang dihadapi.

Sampel yang digunakan dalam penelitian ini adalah 8 mahasiswa yang terdiri dari 7 laki-laki dan 1 perempuan. Sampel diambil berdasarkan tingkat tahapan penyelesaian, di mana diambil sampel dengan tingkatan tahapan penyelesaian masalah yang bervariasi, sehingga informasi mengenai kesulitan mahasiswa dalam penyelesaian masalah lebih lengkap. Berikut analisis pekerjaan dan hasil wawancara tiap subyek penelitian.

Subyek 1

Hasil Program:

Hasil program masih banyak kesalahan sehingga belum dapat dijalankan programnya. Dari hasil program dapat disimpulkan bahwa subyek tidak bisa mengidentifikasi tiap-tiap deskripsi masalah yang diberikan, baik input maupun output yang diminta. Hal ini dapat dilihat dari pemilihan variabel dan tipe data yang salah. Selain itu kemampuan implementasi program juga masih kurang. Hal ini dapat dilihat pada banyaknya kesalahan penulisan sintaks (*illegal expression*), terutama pendeklarasian variabel dan akses nilai variabel yang tidak sesuai dengan tipe datanya. Lebih jelasnya lihat Gambar 1.

```

[1]
program kambing2222;
uses crt;
type Kambing=record
    kandang:char;
    nomor:integer;
    berat:integer;
end;
kKambing= array[1..8] of kKambing;
var A,B,C: kKambing;
    n,k,l,j,x:integer;
begin
clrscr;
write('banyak kandang tiap baris= '); readln(n);
writeLn('Kandang #');
for i:= 1 to n do begin
with B[i] do begin
writeLn(' Nomor : ');
write('Barat : '); readln(berat);
writeLn();
end; end;
end; end;
22:50
  
```

Gambar 1

Hasil wawancara:

Dari hasil wawancara, diperoleh fakta bahwa subyek mampu memahami masalah nyata, tapi kesulitan membawa masalah nyata ke dalam bentuk masalah pemrograman. Subyek tidak bisa menentukan tipe data variabel input, tidak yakin

antara menggunakan array dan record. Hal ini menunjukkan pemahaman yang kurang mengenai tipe data *array* dan *record* sehingga tidak bisa mengasosiasikan tipe data yang tepat terhadap masalah yang diberikan. Dalam hal perencanaan, subyek tidak membuat langkah-langkah secara sistematis, langsung menerapkan dalam bahasa pemrograman. Dalam hal implementasi, subyek masih kesulitan dalam menuliskan sintaks program sehingga masih banyak kesalahan.

Subyek 2

Hasil program:

Pekerjaan telah dicompile dengan baik, tanpa ada kesalahan sintaks. Program meminta masukkan berupa berat kambing dari dua buah kambing tapi tidak sesuai dengan permintaan. Berat setiap kambing dalam kandang A dituliskan secara vertikal sedangkan pada soal horizontal. Selain itu spesifikasi program untuk meminta **inputan banyak kambing pada tiap kandang tidak dipenuhi** dan total berat kambing juga tidak ada.

Berdasarkan program yang sudah dibuat disimpulkan bahwa subyek cukup bisa memahami permasalahan namun kurang detail dalam melihat spesifikasi program yang diinginkan. Subyek sudah mampu membuat rencana penyelesaian, menjalankan rencana, dan juga melihat kembali penyelesaian. Namun demikian pada proses review subyek tidak melihat kembali secara detail mengenai spesifikasi outputan yang diinginkan, berdasarkan contoh pada soal. Lebih jelasnya dapat dilihat pada Gambar 2.



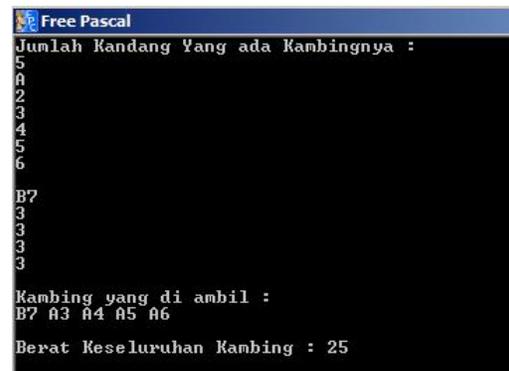
```
Free Pascal
insert berat kambing kandang A
4
5
3
6
7
2
3
4
insert berat kambing kandang B
5
6
7
3
4
1
2
3
B5 B6 B7 A6 A7 A2 A3 A4
```

Gambar 2

Hasil wawancara:

Subyek sudah memahami permasalahan dengan benar, mampu menentukan variabel input beserta tipe data dengan tepat (tidak sesuai dengan hasil program yang dibuat). Pada tahap perencanaan, Subyek membuat langkah-langkah awal (digambarkan dulu) pada kertas sebelum koding. Pada proses memeriksa jawaban, subyek sudah melakukan *compiling*, sehingga program berhasil dijalankan. Namun pada proses ini kurang sempurna karena tidak memeriksa apakah hasil program yang dijalankan sudah sesuai dengan *test case* atau belum.

Subyek 3



```
Free Pascal
Jumlah Kandang Yang ada Kambingnya :
5
0
2
3
4
5
6

B7
3
3
3
3

Kambing yang di ambil :
B7 A3 A4 A5 A6

Berat Keseluruhan Kambing : 25
```

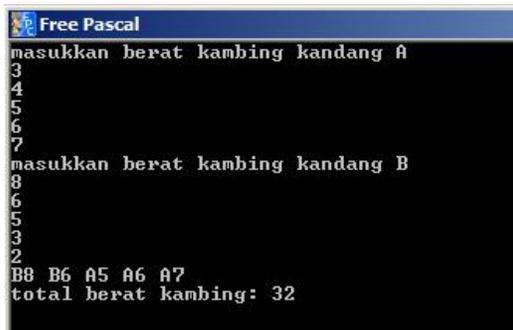
Gambar 3

Hasil pekerjaan : pekerjaan telah dicompile dengan baik. Secara algoritma sudah benar, menampilkan input dan output dengan benar, meskipun secara tampilan tidak sesuai dengan test case yang diberikan (lihat Gambar 3). Dari hasil program

yang dibuat, subyek telah berhasil melaksanakan seluruh tahapan penyelesaian masalah dengan baik. Sedikit kekurangan pada proses melihat kembali pekerjaan yang tidak mendasarkan pada *test case* yang diberikan.

Hasil wawancara: Subyek berhasil memahami permasalahan dengan baik, termasuk dalam menentukan spesifikasi program, input dan output. Pada tahap menyusun rencana subyek langsung menyusun kode program tanpa membuat rencana. Subyek juga melihat kembali pekerjaan namun kurang detail terutama masalah tampilan.

Subyek 4



```
Free Pascal
masukkan berat kambing kandang A
3
4
5
6
7
masukkan berat kambing kandang B
6
5
4
3
2
B6 B6 A5 A6 A7
total berat kambing: 32
```

Gambar 4

Hasil pekerjaan : program dapat di-*compile* tanpa kesalahan, mampu membaca inputan kandang A dan B dan memberikan output dengan benar **tapi tanpa banyaknya kambing dalam tiap kandang** (ditentukan hanya 5 padahal di contoh yang menentukan adalah *user*, lihat Gambar 4). Dari hasil program, subyek kemungkinan kurang seksama dalam memahami permasalahan secara menyeluruh dan dalam proses melihat kembali pekerjaan tidak berdasarkan *test case* yang ditentukan.

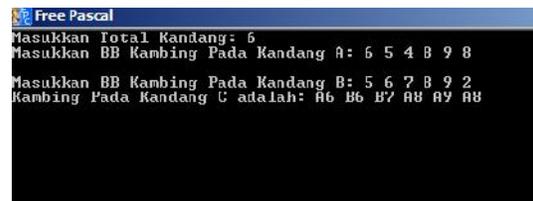
Hasil wawancara:

Subyek sebenarnya memahami permasalahan dengan baik, namun kurang detail dalam membaca spesifikasi program sehingga ada spesifikasi input

yang terlewatkan, yaitu banyak kambing dalam tiap kandang. Selain itu subyek tidak menuliskan rencana terlebih dahulu, langsung menuliskan kode program. Pada tahap melihat kembali pekerjaan, subyek tidak mendasarkan pada *test case* yang idebrikan pada soal sehingga ada satu spesifikasi input yang terlewat.

Subyek 5 (Dewangga)

Hasil program : Sudah sesuai dengan yang ditentukan, namun ada satu spesifikasi output yang belum ditampilkan yaitu total berat kambing yang sudah dipilih, meskipun sudah ada dalam algoritmanya. Kemungkinan kurang teliti dalam melakukan melihat kembali pekerjaan. Untuk lebih jelasnya, lihat Gambar 5.



```
Free Pascal
Masukkan Total Kandang: 6
Masukkan BB Kambing Pada Kandang A: 6 5 4 3 2
Masukkan BB Kambing Pada Kandang B: 5 6 7 8 9 2
Kambing Pada Kandang C adalah: A6 B6 B7 A8 A9 A8
```

Gambar 5

Hasil wawancara:

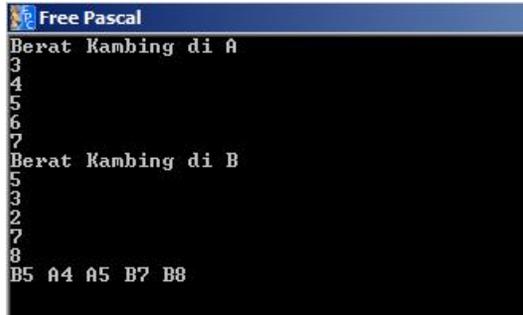
Dari hasil wawancara, subyek telah memahami masalah dengan baik, menentukan spesifikasi program dengan benar termasuk input dan output. Namun dalam menyusun rencana tidak dilakukan secara tertulis, langsung menerapkan pada kode program. Kekurangannya pada tahap melihat kembali pekerjaan, tidak mendasarkan pada *test case* meskipun secara algoritma sudah ada pada program. Alasan yang dikemukakan adalah waktu pengerjaan kurang.

Subyek 6

Hasil program :

program dapat di-*compile* tanpa kesalahan, mampu membaca inputan kandang A dan B dan

memberikan output dengan benar **tapi tanpa banyaknya kambing dalam tiap kandang ditentukan hanya 5 padahal di contoh sampai yang menentukan adalah user).**



```
Free Pascal
Berat Kambing di A
3
4
5
6
7
Berat Kambing di B
5
3
2
7
8
B5 A4 A5 B7 B8
```

Gambar 6

Dari hasil program, subyek kemungkinan kurang seksama dalam memahami permasalahan secara menyeluruh dan dalam proses melihat kembali pekerjaan tidak berdasarkan test case yang ditentukan. Lihat Gambar 6 untuk lebih jelasnya.

Hasil wawancara

Subyek sebenarnya memahami permasalahan dengan baik, namun kurang detail dalam membaca spesifikasi program sehingga ada spesifikasi input yang terlewatkan, yaitu banyak kambing dalam tiap kandang. Selain itu subyek tidak menuliskan rencana terlebih dahulu, langsung menuliskan kode program. Pada tahap melihat kembali pekerjaan, subyek tidak mendasarkan pada test case yang diberikan pada soal sehingga ada satu spesifikasi input yang terlewat.

Subyek 7

Hasil program:

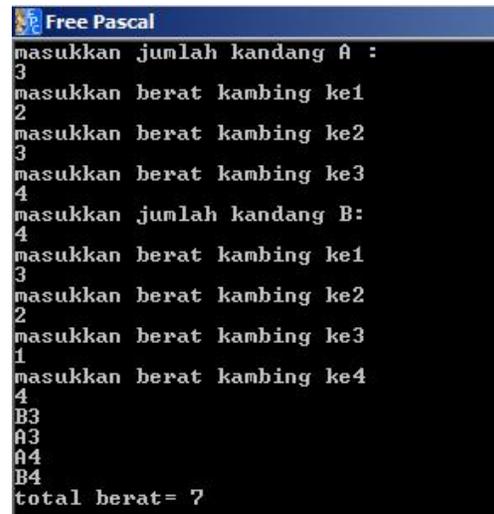
Program berhasil dicompile dan dijalankan. Ditemukan kesalahan, yaitu banyaknya kambing tiap kandang berbeda (lihat Gambar 7). Hal ini tidak sesuai dengan ketentuan bahwa tiap kandang mempunyai banyak kambing yang sama dan dibandingkan satu persatu. Subyek membuat inputan banyak kambing dari tiap kandang

sehingga ada kemungkinan kambing yang dibandingkan tidak sama banyaknya.

Format juga kurang sesuai dengan ketentuan. Dari program yang telah dibuat, disimpulkan subyek kurang bisa memahami soal dengan baik meskipun mampu merencanakan dan menjalankan rencana dengan baik karena sudah terlihat hasilnya.

Hasil wawancara:

Dari hasil wawancara, subyek kurang bisa memahami soal, menentukan spesifikasi program seperti memahami memahami input dan output yang dibutuhkan oleh program. Selain itu subyek juga tidak membuat rencana sebelum menyusun kode program. Pada tahap melihat kembali pekerjaan subyek tidak mendasarkan hasilnya menggunakan contoh test case yang diberikan.



```
Free Pascal
masukkan jumlah kandang A :
3
masukkan berat kambing ke1
2
masukkan berat kambing ke2
3
masukkan berat kambing ke3
4
masukkan jumlah kandang B:
4
masukkan berat kambing ke1
3
masukkan berat kambing ke2
2
masukkan berat kambing ke3
1
masukkan berat kambing ke4
4
B3
A3
A4
B4
total berat= 7
```

Gambar 7

Subyek 8

Hasil program

Program masih ada yang error, tidak bisa dijalankan. Kesalahan yang terlihat dari program yang dibuat adalah dalam penentuan variabel dan tipe data. Kesalahan juga terlihat pada langkah-langkah penyelesaian dan juga dalam penulisan, terlihat masih adanya sintaks yang salah. (lihat Gambar 8.



Gambar 8.

Hasil wawancara:

Subyek kesulitan dalam memodelkan masalah nyata ke dalam bahasa pemrograman. Subyek mengetahui inputan dan outputan dalam kehidupan nyata, namun tidak bisa mengidentifikasi ke dalam variabel dan tipe data yang sesuai. Subyek menyusun rencana penyelesaian dalam bentuk coretan sebelum menyusun kode program, namun tidak dalam bentuk pseudocode atau flowchart.

Dari hasil tes dan wawancara dapat diperoleh kesulitan-kesulitan yang dihadapi oleh mahasiswa dalam penyelesaian masalah pemrograman sesuai dengan 4 tahapan Polya.

a. Memahami permasalahan

- Kesulitan dalam membawa masalah nyata ke dalam bahasa pemrograman, termasuk di antaranya menentukan variabel input/output dan tipe datanya
- Kesulitan dalam mengidentifikasi spesifikasi program yang harus dipenuhi misalkan input dan output yang harus ditampilkan meliputi apa saja dan bagaimana tampilan yang diinginkan. Dari beberapa subyek, banyak program yang tidak lengkap input maupun outputnya, dan ketidak sesuain format tampilan dengan contohnya.

b. Menyusun rencana

- Mahasiswa kesulitan dalam menyusun rencana (algoritma) secara tertulis, baik

dalam bahasa natural, pseudocode maupun flowchart karena tidak terbiasa. Hampir sebagian besar mahasiswa langsung menyusun kode program berdasarkan apa yang dipikirkan, tanpa menuliskan langkah-langkahnya terlebih dahulu

c. Melaksanakan rencana

- Beberapa mahasiswa kesulitan dalam memahami sintaks. Hal ini terlihat masih ada beberapa program yang masih meninggalkan kesalahan sehingga program tidak bisa dijalankan. Sintaks di sini meliputi kurangnya tanda baca, sintaks penulisan variabel, tipe, dan cara akses variabel, khususnya dalam hal ini adalah array dan record.

d. Memeriksa kembali penyelesaian

- Kesulitan dalam memahami sintaks sehingga tidak mampu mengkoreksi kesalahan sintaks. Hal ini terlihat dari masih adanya program yang meninggalkan kesalahan.
- Mahasiswa tidak menggunakan *test case* yang diberikan pada soal sebagai acuan validitas program yang sudah dibuat.

4. KESIMPULAN

Dari pembahasan di atas dapat disimpulkan kesulitan mahasiswa dalam penyelesaian masalah menurut Polya adalah sebagai berikut:

a. Memahami permasalahan

- Kesulitan dalam membawa masalah nyata ke dalam bahasa pemrograman, termasuk di antaranya menentukan variabel input/output dan tipe datanya
- Kesulitan dalam mengidentifikasi spesifikasi program yang harus dipenuhi misalkan input dan output yang harus ditampilkan meliputi apa saja dan bagaimana tampilan yang diinginkan.

- b. Menyusun rencana
- c. Mahasiswa kesulitan dalam menyusun rencana (algoritma) secara tertulis, baik dalam bahasa natural, pseudocode maupun flowchart karena tidak terbiasa. Melaksanakan rencana
- Beberapa mahasiswa kesulitan dalam memahami sintaks. Hal ini terlihat masih ada beberapa program yang masih meninggalkan kesalahan sehingga program tidak bisa dijalankan. Sintaks di sini meliputi kurangnya tanda baca, sintaks penulisan variabel, tipe, dan cara akses variabel, khususnya dalam hal ini adalah array dan record.
- d. Memeriksa kembali penyelesaian
- Kesulitan dalam memahami sintaks sehingga tidak mampu mengkoreksi kesalahan sintaks.
 - Mahasiswa tidak menggunakan *test case* yang diberikan pada soal sebagai acuan validitas program yang sudah dibuat.

Berdasarkan hasil di atas, perlu ditingkatkan kemampuan pemecahan masalah terutama pada tahap memahami masalah, terutama membawa masalah nyata ke dalam bahasa pemrograman, menentukan variabel input/output dan tipe datanya Dan mengidentifikasi spesifikasi program yang harus dipenuhi. Salah satu caranya adalah dengan membiasakan mahasiswa menghadapi permasalahan-permasalahan nyata yang dapat diselesaikan dengan pemrograman.

Daftar Pustaka

- Barnes, D.J., Fincher, S. & Thompson S. (1997) *Introductory problem Solving in Computer Science*. In G. Daughton & P. Magee(Eds.)5th Annual Conference on the Teaching of Computing (pp.36-39). Centre for Teaching Computing. Dublin City University, Dublin, Ireland
- https://www.tutorialspoint.com/pascal/pascal_tutorial.pdf, diakses tanggal 10 Nopember 2016, jam 13.50.
- McIntosh, R. & Jarret, D. (2000). *Teaching mathematical problem solving: Implementing the vision*. New York: NWREL, Mathematics and Science Education Center.
- Polya, (1957). *How to Solve It.*, 2nd ed., Princeton University Press, ISBN 0-691-08097-6. UNDERSTANDING THE PROBLEM
- Voskoglou, M. Gr., Buckley Sheryl. (2012). *Problem Solving and Computers in a Learning Environment*. Egyptian Computer Science Journal, ECS, Vol.36 No4. ISSN-1110-2586