KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
https://jti.polinema.ac.id

Mata Kuliah	:	Bahasa Inggris 2
Program Studi	:	D4 – Teknik Informatika
Semester	:	3

Kelas	:	TI - 2A
NIM	:	244107020171
Nama	:	Indhira Yuantika Christy
Pertemuan Ke-	:	1

UNIT 1 : PROGRAMMING

Exercise 1 : Have you ever seen any programming code? Here is an example of a C program. What does the program tell you? .

```
#include <stdio.h>
main ()
{

Printf("good
morning\n");

}
```

- Yes, I have seen programming cede before. The program tells the computer to display the message “good morning” on the screen.

Then, discuss with your partner what you think programming is and would you like to be good at programming.

Look at the section of code and the explanations and answer these questions.

1. Find an example of a constant in the code.
   - The example of constant in the code are 3 and 2.
2. What do you think the value of x is, after the third instruction?
   - After third instruction value of x is 5, because x is calculated from variable a (3), and variable b (2). When we add 3 and 2 the result is 5.

Exercise 4: Match the words 1-5 with the definition (A-E).

1. Flowchart : E. a diagram representing the successful logical steps of the program.

2. Source Code : A. program instructions written in a particular computer language

3. Compiler : G. a special program which converts the source program into machine code-the only language understood by the processor.

4. Machine Code : H. the basic instructions understood by computers, consisting of 1s and 0s (binary code).

5. Debugging : C. the techniques of detecting and correcting errors which may occur in programs

6. Assembly Language : F. Low-level language translated into machine code

7. High-level Language : D. programming languages such as C, Java, or Visual Basic

8. Markup Language : B. language used to create and format documents for the Web

Exercise 7: Fill the missing words to complete the text. Use the words in the box.

To write a (1) program software engineers usually follow these steps. First, they try to understand the problem and define the purpose of the program. Next, they design a step-by step plan of instructions. This usually takes the form of a (2) flowchart, a diagram that uses standardized symbols showing the logical relationship between the various parts of the program. These logical steps are then translated into instructions written in a high-level computer (3) language (PASCAL, COBOL, C++, etc.). These computer instructions are called the 'source code'. The program is then (4) compiled, a process that converts the source code into machine code (binary code), the language that computers understand. Testing program are then run to detect (5) errors in the program. Errors are known as 'bugs', and the process of correcting these errors is called (6) debugging. Engineers must find the origin of each error, then write the correct instruction, compile the program again, and conduct another series of tests. Debugging continues until the program runs smoothly. Finally, software developers write detailed (7) documentation for the users. Manuals tell us how to use programs like word processors, databases, or web browsers.

Exercise 8:

Steps in Programming

The first step in programming is to understand the problem to be solved. After that, determine the required input data and the desired output. The next step is to develop rules and a solution plan in the form of an algorithm, which is a sequence of logical steps to solve the problem. Algorithms are usually written in flowcharts to make them easier to understand. Once the plan is clear, the program can be written using a programming language. After that, the program will be tested to ensure that the results are correct. If there are errors, corrections will be made. Once it has been confirmed that the program is correct, it is complete.

Exercise 9: Programmers sometimes use flowchart when they are planning a program. These following symbols are used in making flowchart. Identify each and its function.

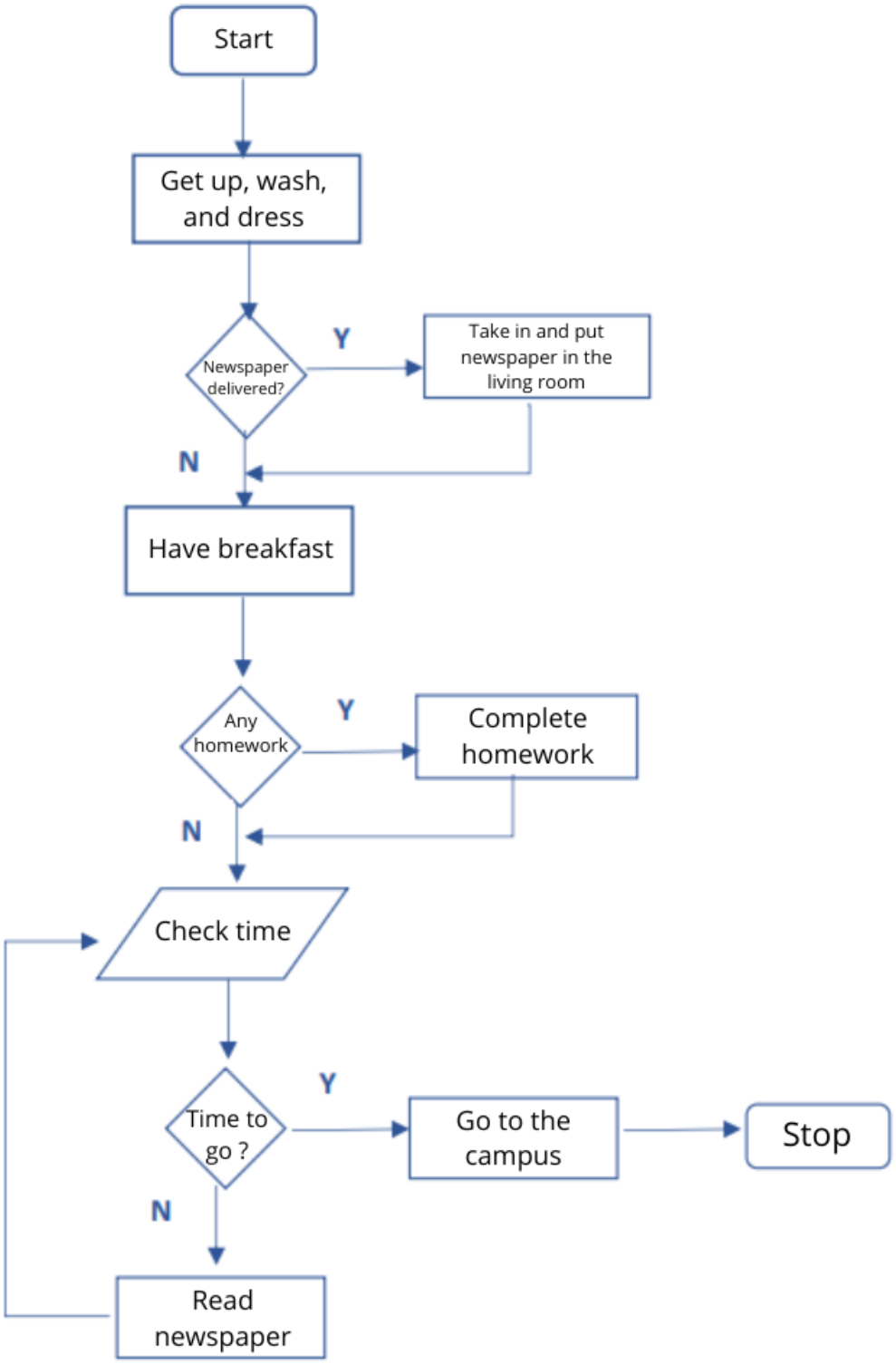
No.	Symbols	Names	Functions
1.		Terminal Start/end	Indicates the beginning or the end of a program or process.
2.		Input/Output	Represents input (data entry) or output
3.		Process	Represents a process, action, or instruction to be executed.
4.		Decision	Represents a decision point in the program, usually with Yes/No answers.
5.		Flowline	Shows the direction of the flow or sequence of steps in the process.
6.		Connector	Connects different parts of a flowchart, often across pages.
7.		Document	Represents a document or printed output generated by the process.

Exercise 10: Read this text carefully and then do the exercises.

Decide whether the following statements are true (T) or false (F). Then make the necessary changes so that false statements become true.

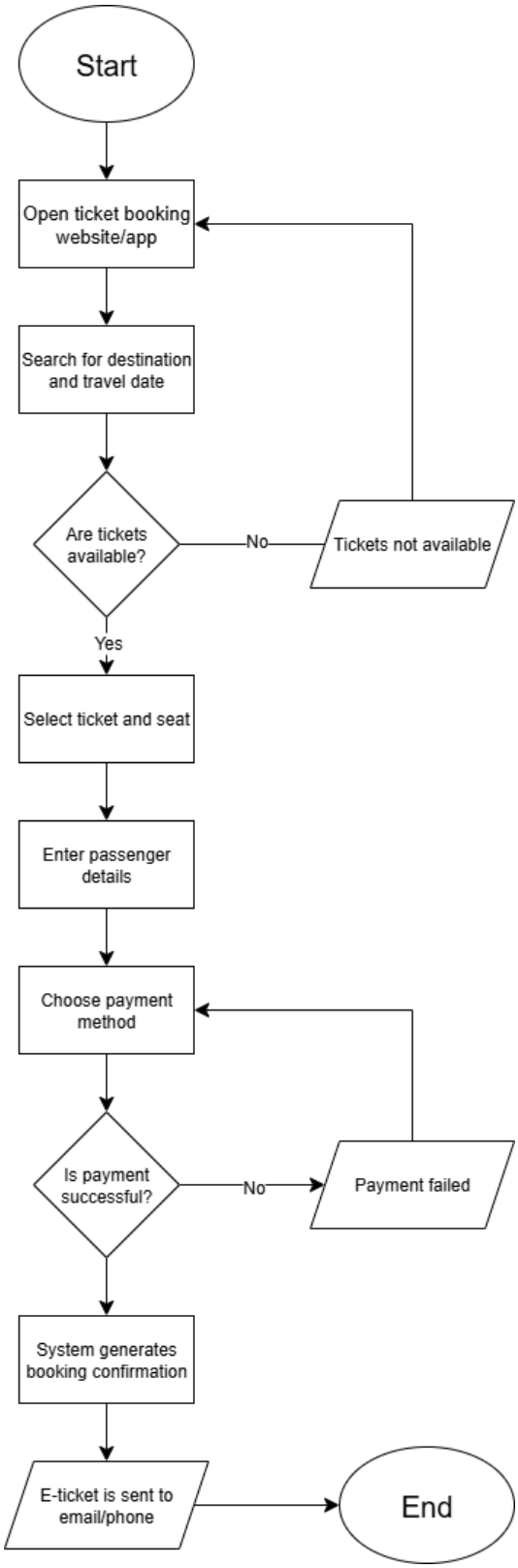
1. A good flowchart takes into account the steps which are necessary to solve the problem.
   - True
2. It is not possible to draw a flowchart without using a template.
   - False: It is possible to draw a flowchart freehand without using a template.
3. There is only one possible flowchart for every problem.
   - False: There can be different flowchart for the same problem.
4. Every programmer must learn flowcharting and realize its importance.
   - True
5. The method of flowcharting depends on the programming language being used.
   - False: Flowcharting does not depend on the programming language.
6. Flowcharts show the logic one has to follow to solve a problem.
   - True
7. Documenting a program is essential in explaining what the program is supposed to do.
   - True
8. If the flowchart is correct, the program will work.
   - False: Even if the flowchart is correct, the program still need to be tesed to make sure it works.
9. Each symbol in flowcharting has a specific meaning.
   - True
10. Flowcharts can show processes, but not decisions.
    - False: Flowchart can show both processes and decisions.

Exercise 11: Flowchart David's activities by completing the flowchart below.



Exercise 12: Draw a flowchart for one of these activities.

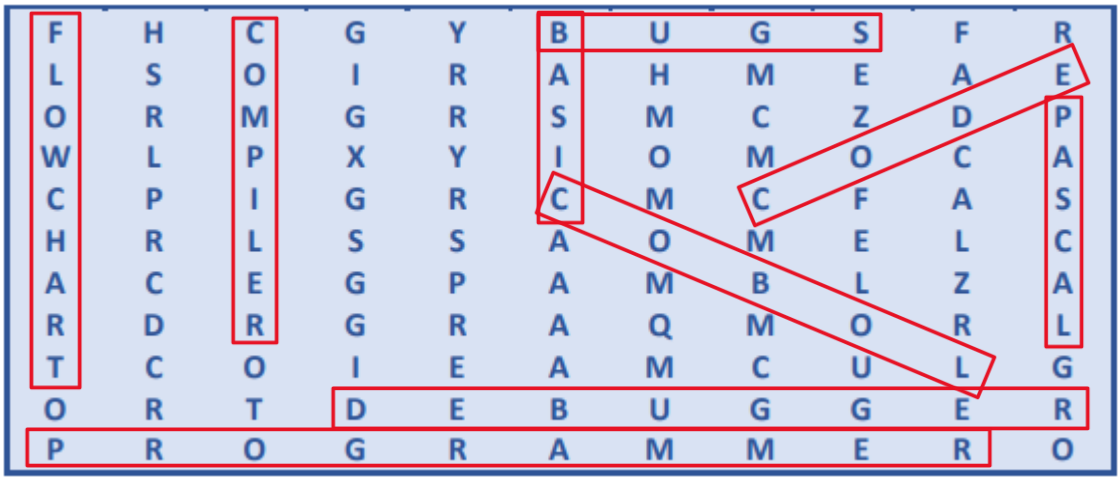
6. Booking a ticket online.



Exercise 13: Now, write the description of the flowchart you have made on Exercise 10 and present it in front of the class.

- This flowchart illustrates the process of booking a ticket online. The process starts when the user opens the ticket booking website or application. The user searches for the destinations and travel date. If no tickets are available, the system display the message “Ticket not available and the user opens another application or website to continue searching. If tickets are available, the user selects the ticket and seat, then enters the passenger details. After that, the user chooses a payment method. If the payment fails, the system displays the message “Payment failed”, and the user is redirected back to choose another payment method until the payment is successful. Once the payment is successful, the system generates a booking confirmation and sends the e-ticket to the user’s email or phone. Finally, the process ends.

Exercise 14: Find 10 words about Programming.



1. Compiler
2. Flowchart
3. Debugger
4. Programmer
5. Pascal
6. Bugs
7. RAM
8. Code
9. Basic
10. COBOL

Exercise 15: Read the text carefully and answer the following questions.

Answer the questions briefly.

1. Do computers understand human languages? Why/Why not?
    - No, computers don't understand human languages. The only language they can understand directly is machine code.
2. What is the function of an assembler?
    - An assembles is piece of software that translates a program from an assembly language into machine code.
3. How many high-level languages are mentioned? What are they?
    - There are 7 high level languages, there are FORTRAN, COBOL, BASIC, PASCAL, C, C++, and JAVA
4. Why did software developers design high-level languages?
    - Software developers design high-level language to make the programs easier to write and to overcome the problem of intercommunication between different types of computers. They are closer to the English language.
5. What is the difference between a compiler and an interpreter?
    - The difference between a compiler and an interpreter in that a compiler translates the entire source code into machine code all at once, while an interpreter translates and executes the source code line by line as the program is running.
6. Why are HTML and VoiceXML called markup languages?
    - HTML and VoiceXML are called markup language because they use instructions, known as markup tags, to format and link text files, rather than being used for programming.

Exercise 16: Complete these sentences with a computer language from the text.

1. XML allows us to create our own tags to describe our data better. We aren't constrained by a pre-defined set of tags the way we are with HTML.

2. IBM developed FORTRAN in the 1950s. It was the first high-level language in data processing.

3. JAVA applets are small programs that run automatically on web pages and let you watch animated characters, play games, etc.

4. VoiceXML is the HTML of the voice web. Instead of using a web browser and a keyboard, you interact with a voice browser by listening to pre-recorded audio output and sending audio input through a telephone.

5. This language is widely used in the business community. For example, the statement ADD VAT to NET-PRICE could be used in a COBOL program.

Exercise 17: Report each of these screen messages.

1. Make sure the printer is switched on before continuing.
   - It tells you to make sure the printer is switched on before continuing.
2. Game mode is on.
   - It informs you that the Game Mode is on.
3. Do you want to create a new document?
   - It asks you if/whether you want to create a new document.
4. What is the captcha code?
   - It asks you what the captcha code is.
5. Fill in your name in the box.
   - It requires you to fill in your name in the box.
6. Please type the next number.
   - It requires you to type the next number.
7. Enter your password.
   - It requires you to enter your password.
8. Please choose from menu below.
   - It requires you to choose from the menu below.
9. Can't rename "Pictures" because a folder with that name already exists.
   - It informs you that it can't rename "Picture" because a folder with name already exists.
10. Exit?
    - It asks you if/whether you want to exit.
11. Are you sure you want to copy the selected files?
    - It asks you if/ whether you are sure you want to copy the selected files.
12. Do you want to defrag the drive?
    - It asks you if/whether you want to defrag the drive.
13. Mute story and posts?
    - It asks you if/whether you want to mute stories and posts.
14. If you unfollow this account, you'll have to request to follow again.
    - It informs you that if you unfollow this account, you will have to request to follow again.
15. Click the subscribe button to follow us.
    - It requires you to click the subscribe button to follow us.