Automated Study Plan Generator using Rule-based and Knapsack Problem

Muhammad Amin Mustapa, Lizawati Salahuddin, Ummi Rabaah Hashim
Fakulti Teknologi Maklumat dan Komunikasi,
Universiti Teknikal Malaysia Melaka (UTeM)
Durian Tunggal, Melaka, Malaysia

***Abstract*—Undergraduate students are given the flexibility of arranging courses throughout their study duration especially when they are eligible for credit exemption for the courses taken during their diploma study. Issues arise when students arrange their studies manually. Improper course arrangement in the study plan may be resulting some of the selected courses do not correspond to the courses offered, and imbalance credit hours. Hence, this study aims to propose an algorithm to generate an automated and accurate study plan throughout the study duration. A combination of rule-based and knapsack problem were proposed to generate an automated study plan. A quantitative methodology through expert's reviews and questionnaire survey was conducted to evaluate the accuracy of the proposed algorithm. The proposed algorithm shows high accuracy. In conclusion, the combination of rule-based and knapsack problem is appropriate to generate an automated and accurate study plan. The automated study plan generator can help students generate an effective study plan.**

***Keywords*—Knapsack problem; rule-based; study plan; undergraduate; credit exemption**

I. Introduction

Study planning is important to ensure the students carry a balance study load in every semester. The balance of courses and the number of credit hours chosen by the students themselves determine the planning of non-burdensome study sessions. Students need to allocate time (also known as student learning hours) for the implementation of all learning activities to achieve the learning outcomes. The student learning hours includes formal meetings (e.g., lectures), guided learning (e.g., tutorials, seminars, internships, and fieldwork), self-directed learning, and preparations for tests and final exams. A balance study load could influence the student's academic performance. Study planning is becoming more critical for undergraduate students who are eligible for credit exemption for the courses taken during their diploma study. When each student has a different number of total credit exemptions, the difference becomes more pronounced. As a result, different study plan is devised for each student. With a well study plan, students may shorten their undergraduate study depending on the total credit exemption.

Study planning is tied to academic rules. For instance, students are allowed to take minimum of 12 credits (exception on the last semester of study), and maximum of 20 credits per semester. Besides, a prerequisite course must be completed prior to another course. Moreover, not all courses are available in every semester. Some courses are only available in odd semesters, while others are only available in even semesters. Therefore, courses should be arranged according to the curriculum structure, and total credit hours per semester should be divided appropriately. The study plan should not interfere with the learning journey.

The field of artificial intelligence (AI) and knowledge-based system has enormous potential for improving simulation modelling support [1, 2]. Due to recent advances in the field of AI, a knowledge-based system has demonstrated its abilities by providing successful solutions in a wide range of applications including in the field of education [2], agriculture [3], manufacturing [4], and health [5]. The system can be used as an alternative to traditional systems, particularly in advisory tasks and symbolic reasoning [6]. It is a subfield of AI that collects data automatically without the assistance of a human expert to solve problems that normally necessitate human intelligence [7, 8].

Rules can be viewed as a simulation of the cognitive behavior of human experts. A rule-based expert system can mimic the ability of human experts to make decisions [9], [10]. They are programmed to solve problems in the same way that humans do, by using stored human information or expertise. Rule-based structures are created to solve specific problems in a given domain. Every domain has its own set of intelligent and reasoning humans that can be modelled and even replaced by automated rule-based systems. A system generator based on a rule engine that uses an improved Rete algorithm was designed to match data objects to perform certain functions through a system generator using rules set by the user (production) [11]. The rule engine's primary responsibility is to match the data objects submitted to the engine with the business rules, activate the business rules based on the current data state, and trigger the operations in the application based on the execution of logic declared in the ruleset. Reference [12] states that the problem of scheduling by minimizing the amount of flow time has attracted more attention from the research community. This is because the lower the total flow time value, the greater the resource utilization and cost savings. In this regard, today's manufacturing environment is quite practical, as it reduces the amount of flow time. Several tasks comprised of some sequences are utilized to determine optimal values for minimizing overall flow time; to provide good solutions as the problem size expands the development of heuristics and meta-heuristics is essential. In the study, a ruled-based heuristic process for determining the sequence with the least total flow time is proposed. The experimental results show that the proposed approach makes a major contribution to the exceedingly difficult scheduling problem.

The basic principle of all knapsack problem families is to choose a few objects, each with a benefit and weight value, to be packed into one or more capacity knapsacks. Assume there is a group of elements with known weights and values, as well as a pack or bag with a limited capacity for filling the knapsack. A problem known as the knapsack problem is devised to fill the said pack with the elements in such a way that their aggregate sum is possibly the highest without exceeding the pack's ability [13]. Knapsack Problem 0-1 is a popular form of knapsack problem with a wide range of applications, including capital budgeting, project selection, resource allocation, cutting stock, and investment decision-making. As a result, the issue of Knapsack Problem 0-1 optimization has drawn the attention of an increasing number of researchers [14]. GRASP technique was applied to a nurse-scheduling problem where the goal is to optimize a collection of preferred courses to a set of binding constraints [15]. A critical challenge is striking a balance between feasibility and optimality. Construction heuristics, neighborhood search methods, and evolutionary algorithms have all been effectively utilized to solve real scheduling issues. However, there is a frequent conflict between feasibility and solution quality, as well as difficulties in maintaining an appropriate balance between goals. This is solved by employing a knapsack problem, which ensures that the solutions generated by the construction heuristic are simple to fix. A diversification approach and a dynamic assessment criterion improve the optimum combo even further.

Study in [16] developed an automatic course planning system by using ontology and rule-based. The aim was to create a suitable course plan for a group of students according to the course prerequisite requirement, complexity of the course, teaching method, and the duration of the course. However, the course planning system did not include the course scheduling for a complete study duration from year one until end of study duration. Machine learning techniques were used to group students into similar study pattern according to the CGPA achievement and subsequently determine a feasible study path for the forthcoming semester [17]. Specifically, Neural Network algorithm is used for creating CGPA prediction models, and K-means algorithm is applied to group students according to the similarities of their grades in each course. The evaluation of the proposed system revealed that the students have improved their study performance for their ultimate CGPA in graduation. However, the proposed system does not consider the duration of the study completion, the course prerequisite requirement, and total number of credits in a semester. Moreover, [18] in their research work addressed the issue of determining the ideal set of courses to provide students with in a particular semester, while taking into account the required courses and the availability of teachers to teach those courses. The use of *CourseScheduler*, *IApplet* and *AdmValidatorApplet* function altogether helps the authors to achieve their aim successfully. However, the research focused on the creating a schedule of classes that aid the department administrative in the course scheduling rather than the study plan for students. The method to assist students generating study plan is lacking. Based on these limitations, this study aims to propose and validate an algorithm for compiling study plans throughout the study duration. A more in-depth investigation was conducted to assess the method's accuracy and usefulness.

II. Background

A direct entry student is defined as a student who pursues a degree from a particular institution or a university with a particular completed diploma degree. Compared to direct admission students, direct entry students are allowed to make credit exemption. Credit exemption is a provision of the academic regulations under the semester system that aims to facilitate student mobility. For an instance, students must complete a diploma with at least a 3.00 CGPA from an institution and the courses pursued must be recognized by the senate as equivalent and meet the curriculum requirements of the program pursued or in their respective field of study. Credit exemption may be granted to students who have taken equivalent courses and passed with a minimum grade of C using the university's grading system, provided that at least 80% of the learning content is equivalent. The amount of credit exemption allowed should not exceed 30% of the total credits of the graduating requirements.

Academic handbooks have become a reference for students, containing important information about students' curriculum structure according to a specific program. Courses are divided into four categories, namely general module (W), core module (P), specialization module (K), and free module (E). Table I explains the course category. All courses are categorized as W and are not allowed for credit exemption.

TABLE I. Course Category

Component	**Code**	**Meaning**	**Credits**
General Module	W	University Compulsory Courses, which are a group of important Courses determined by the Senate and made compulsories for all students.	14
Program Core Module	P	Mandatory courses to meet the requirement of Bachelor of Computer Science.	45
Final Year Project			6
Industrial Training			12
Specialization Module	K	Specialization courses to a specific major of an academic program.	30
Free Module	E	Elective Courses that are offered to deepen an academic program.	13
Total Credits			120

TABLE II. COURSE CODE AND NAME

Code	Course Code	Course
P1	BITU 2913	Workshop I
P2	BITU 3973	Final Year Project I
P3	BITU 3983	Final Year Project II
P4	BITP 1113	Programming Technique
P5	BITI 1113	Artificial Intelligence
P6	BITS 1313	Data Communication and Networking
P7	BITP 3113	Object Oriented Programming
P8	BITP 2213	Software Engineering
K1	BITU 3923	Workshop II
K2	BITI 2213	Knowledge Based System
K3	BITI 3413	Natural Language Processing
K4	BITI 2223	Machine Learning
W1	BLHW 1442	English for Academic Purposes
W2	BLHW 2452	Academic Writing
W3	BLHW 3462	English for Professional Interaction
W4	BKK ---1	Co-Curriculum I
W5	BKK ---1	Co-Curriculum II
E1	BLHC 4302	Critical and Creative Thinking

Table II shows the list of course code and name. Students must meet all components of the code to complete a total of 120 credit hours.

III. ALGORITHM IMPLEMENTATION INTO UNIVERSITY RULE

Rule-based expert system is based on knowledge that collects a range of factual information, and makes actions through interpretation from a set of predefined rules [19]. Certain courses have rules that must be followed to complete the semester. According to the rules, the proposed algorithm will decide whether or not to include the course. For the arrangement of study plans, a new rule is proposed to control the arrangement of schedules according to the selected semester. The selection of courses is based on the current semester offers and availability through the knapsack problem method until the credit hour rate reaches a predetermined limit.

Normal students must complete all 120 credits in a minimum of 7 semesters. However, direct entry students have the option to shorten the semester depending on the total number of credit exemption approved. Alternatively, students can stay with 7 semesters as offered with lower credit hours. The first step is to determine the maximum number of semesters according to the total number of credit exemption approved. Then, the total credit hour in a semester is calculated to ensure a balance credit taken by students in every semester, and to ensure that the total credit hours to be taken do not exceed the stipulated conditions. Fig. 1 depicts the semester calculation step as well as the total credit hours in a semester according to total credit exemption approved.

Equation (1) shows the calculation of total credit hours in a semester (*tch*).

$$tch = \frac{120-\mathrm{tce}-12}{(\mathrm{ts}-1)} \qquad (1)$$

where 120 is the minimum graduating credit, *tce* is total credit exemption, 12 is the Industrial Training credit, and *ts* is the total number of semesters.

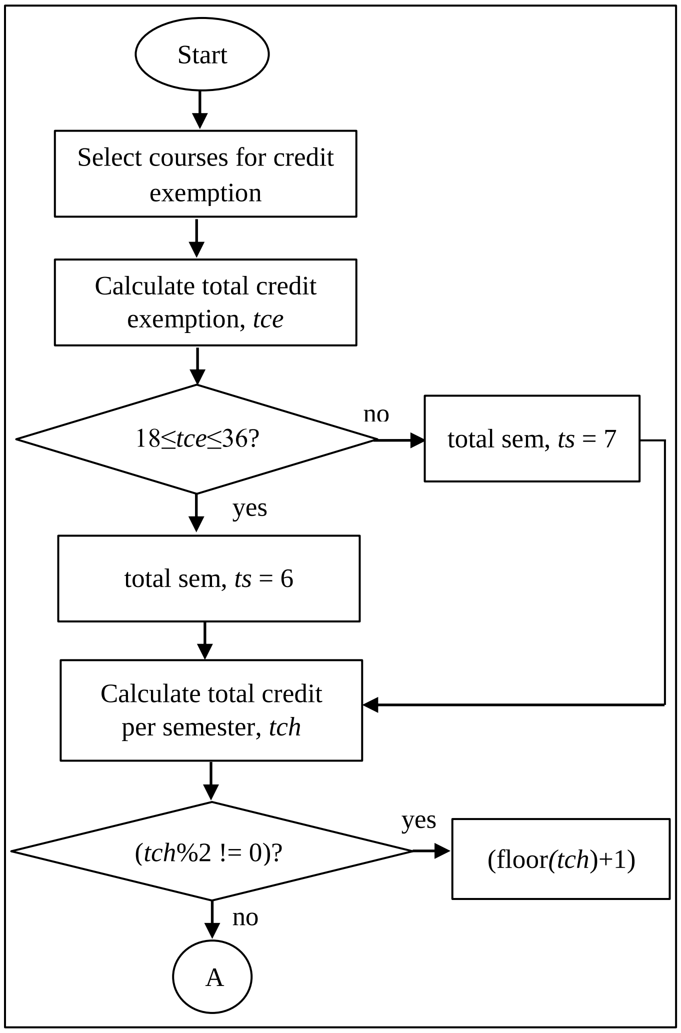

Start
Select courses for credit exemption
Calculate total credit exemption, tce
18≤tce≤36?
no
total sem, ts = 7
yes
total sem, ts = 6
Calculate total credit per semester, tch
(tch%2 != 0)?
yes
(floor(tch)+1)
no
A

Fig. 1. Total Semester and Credit Hour according to Total Credit Exemption.

Industrial Training is a mandatory requirement for students at the end of the semester before graduation. Hence, the industrial training credit and the semester are deducted to calculate *tch*. To make the calculation method simpler, the maximum amount generated will be added to a value of 1 for the semester when the calculation result produces a decimal number and the decimal part is removed as shown in Fig. 1. This is because the number of available credits offered varies and there are no credit hours in decimal form. Using this formula, the total credit hours will not fall below the semester's minimum total credit of 9 and will not exceed the semester's maximum total credit of 20.

Credit exemption is permitted for program core courses. Not all courses can be exempted, including those courses with W category. Workshop I (P1), Workshop II (K1), Final Year Project I (P2), and Final Year Project II (P3) are project-based courses that cannot be exempted. P1 and P2 is the prerequisite course of K1 and P3, respectively. Moreover, K1 is the prerequisite course of P2. As they are offered once a semester and have pre-requisites, these courses should not be taken lightly. The proposed algorithm has set some rules based on the number of semesters. Each rule has unique characteristics for each course. The course is thus removed from the list of available courses because it has become a rule that must be followed. Using the proposed rule-based approach, several courses must be prioritized to ensure the planned flow runs smoothly. Overall, rule-based algorithm is applied to:

- determine number of semesters study
- prioritize University Compulsory Courses to be arranged in the semester according to the program curriculum structure.
- prioritize program core courses according to the pre-requisite and semester offered.
- prioritize specialization courses according to the pre-requisite and semester offered
- prioritize English courses according to the pre-requisite and semester offered.

The parameters used for configuring the rule-based algorithm include the total credit hours, exempted courses and their credit hours, course prerequisites, program curriculum structure and course details. The course details including course name, course code, course category, credit hours, and the semester offered according to odd or even semesters.

A. General Rule for Seven Semesters of Study

Students who only receive credit exemption ranging from 3 to 15 credit hours are advised to complete seven semesters of study. This is because the number of exemption hours is insufficient to reduce the study time. However, students can reduce the credit hours for the coming semesters. The rule prioritizes the courses categorized as W and K to be arranged in the semester according to the curriculum structure as depicted in Fig. 2.

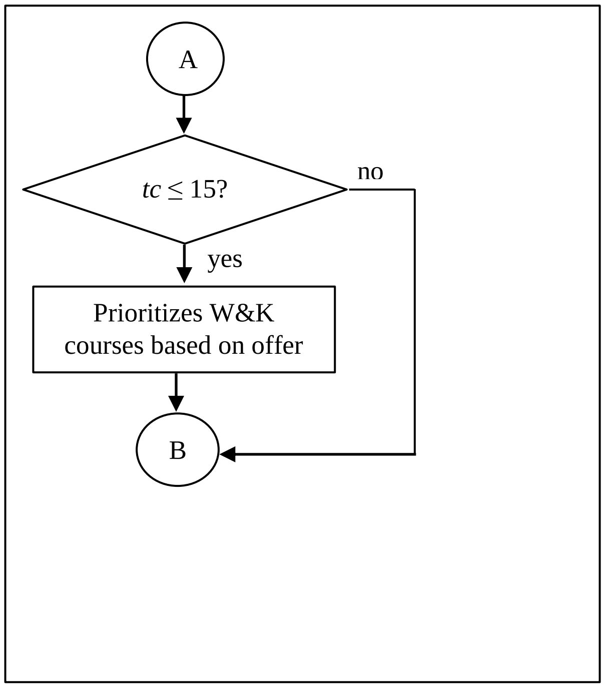


Fig. 2. 7-Semester Rules.

B. General Rule for Six Semesters of Study

Fig. 3 illustrates how the rules for 6 semesters are applied. For all programs offered at the faculty, workshops (P1 and K1) are the main course at the core of the program. The P1 course is available in both semesters, but the K1 course is only offered in the odd semester. Therefore, students are encouraged to take P1 early in the semester so that they can enroll K1 in the subsequent odd semester with 3rd year students. Then, students are allowed to take final project courses in the following semester. Moreover, P4 (Programming Technique) is the pre-requisite course of P1. In this way, planning to shorten the semester is more structured because the core courses can be enrolled in the appropriate semester.

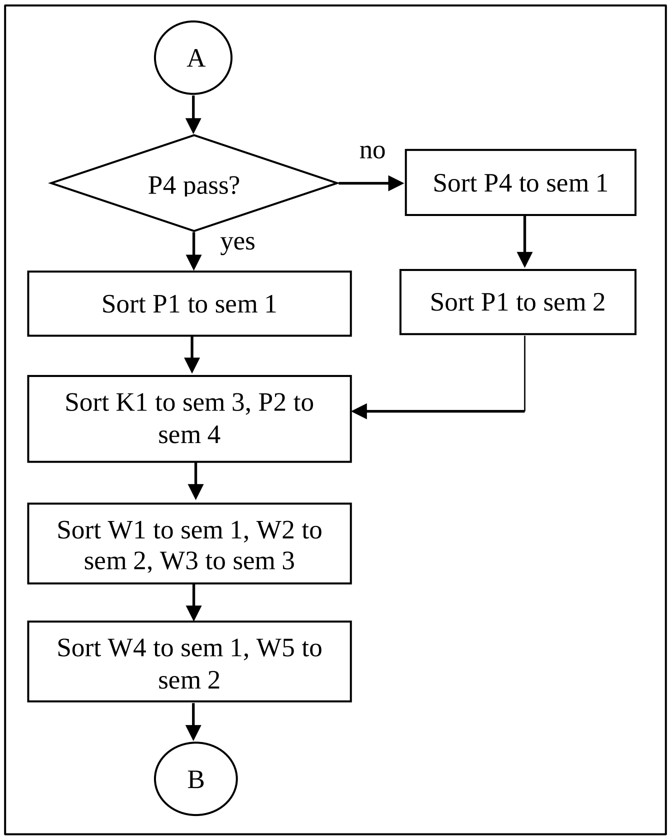

A
P4 pass?
no
Sort P4 to sem 1
yes
Sort P1 to sem 1
Sort P1 to sem 2
Sort K1 to sem 3, P2 to sem 4
Sort W1 to sem 1, W2 to sem 2, W3 to sem 3
Sort W4 to sem 1, W5 to sem 2
B

Fig. 3. 6-Semester Rules.

P1 and K1 are given priority because they are the backbone of the study plan and are divided into six semesters. According to the curriculum structure, these workshop courses are only offered in odd semesters. In semester 1, the algorithm will check the pre-condition status of the P1 course first before allocating the course in a semester. If the requirements are not complied with, students will first consider the P4 and change P1 to the second semester. The group for K1 comprises direct entry students and normal entry students. Hence, the course must be offered in semester 3 to ensure the direct entry students can be assigned in groups. This is equivalent to semester 5 of normal students. Next, English courses are placed in the earlier semester such as English for Academic Purpose (W1), sorted to semester 1, English for Academic Purposes (W2), sorted to semester 2, and English for

Professional Interaction (W3) at semester 3. The next rule ensures the selection of co-curriculum courses. The method is the same as the prerequisites by ensuring that co-curriculum courses are not taken in the same semester and Co-Curriculum II (W5) does not precede Co-Curriculum I (W4).

C. Course Specialization Rules

In addition, specialization based on the program taken by the students is emphasized. This is because these courses are only concentrated among the same programs. The students are not permitted to join specialization classes of other programs. Since specialization courses are offered at a particular semester, a rule is made to allow and ensure specialization courses are taken during the semester where the courses are offered. This ensures the students are following the correct guidelines throughout their study. The rules have limited the students to take 3 or 4 courses per semester to ensure that their study schedule is bearable during the semester. Therefore, maximum specialization courses are set based on the proposed rules. There are several additions to the rules for certain programs such as Bachelor of Computer Science (Database Management) with honors (BITD), Bachelor of Computer Science (Computer Networking) with honors (BITC), and Bachelor of Computer Science (Artificial Intelligence) with honors (BITI). For instance, for the BITI program, the P5 (Artificial Intelligence) course is a prerequisite that must be met. The P5 course affects other courses like K2 (Knowledge-Based System), K3 (Natural Language Processing), and K4 (Machine Learning). Fig. 4 illustrates the additional rules of BITI program.

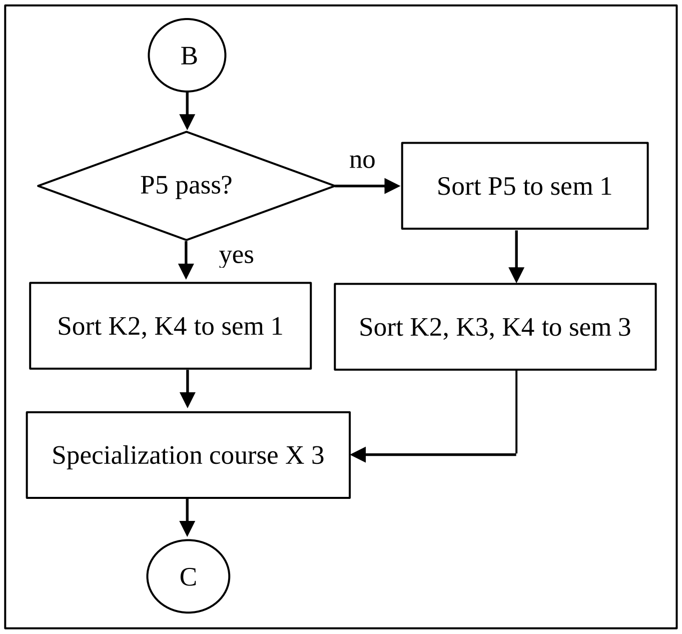


Fig. 4. BITI Specialiazation Rules.

D. Course Availability Rule

The availability of a course should be considered before carrying out this proposed rule. This is because specialization courses need to be sorted accordingly. It is important to offer the courses according to odd or even semesters so that students are not left behind when the courses are offered. If the courses are only offered in odd semesters, they will not be available in even semesters, and vice versa. Some courses are open to other programs in other semesters. Students can plan ahead of time to enter the classes indirectly. This arrangement is based on the lean and the year of the offers to correspond to the students' year of study. This arrangement must be made to ensure that students take a diverse range of courses while also meeting the required credit hours. The proposed algorithm will ensure the availability of a course's semester whether it is in an even or odd semester only or both.

Fig. 5 shows a continuation of the previous compilation of rules. The next rule stipulates that the specialization courses should be included in a particular semester. This is because the semester arrangement is short, and some courses need to be taken first. Specialization courses according to a particular program are usually not offered in other programs; hence should be prioritized in the compilation.

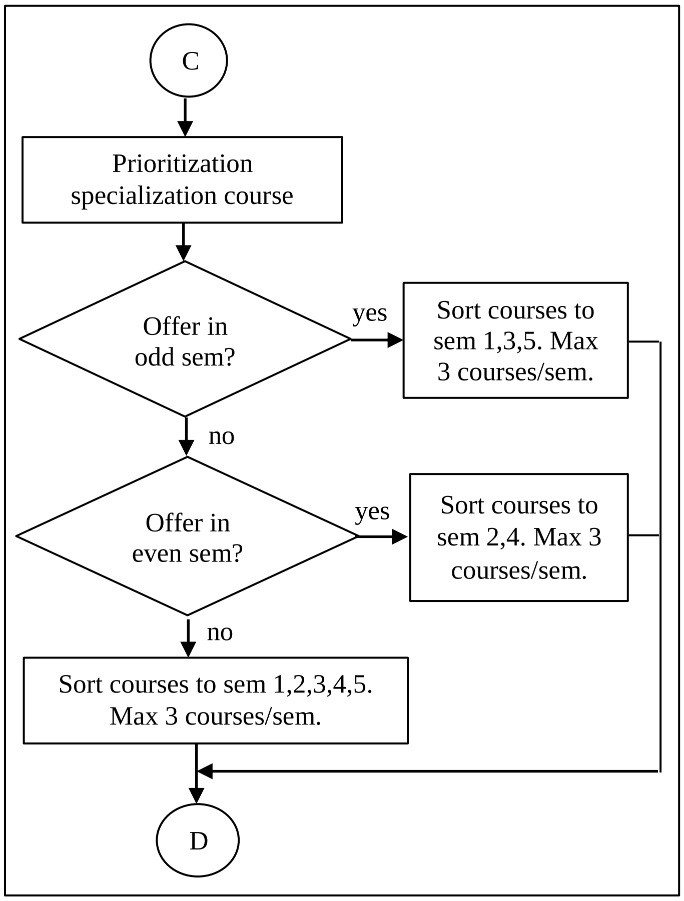

C
Prioritization specialization course
Offer in odd sem?
yes
Sort courses to sem 1,3,5. Max 3 courses/sem.
no
Offer in even sem?
yes
Sort courses to sem 2,4. Max 3 courses/sem.
no
Sort courses to sem 1,2,3,4,5. Max 3 courses/sem.
D

Fig. 5. Specialization Course Flow.

E. Sorting Courses using Knapsack Problem 0-1

The Knapsack Problem 0-1 is applied to fill the remaining number of credit hours from the rule-based algorithm until the total credit hour limit is reached. Equation (2) shows the equation for the Knapsack Problem 0-1.

$$max \sum_{i=1}^{n} xi * pi$$

$$\sum wi\, xi \ \leq c$$

$$xi \ \in \{0,1\}, i = 1, \ldots\ldots, n.$$

$$pi > 0, wi > 0, c > \qquad (2)$$

where i represents course (xi = 1 for selected course, whereas xi = 0 for unselected course), n is a number of total courses, wi is weight, pi is profit which is the credit hour of a particular course, and c is the required remaining credit hours to fulfil the total credit hours per semester. The algorithm will select the highest and most appropriate credit hours that can be adjusted for the number of credit hours remaining. The election results made by the proposed algorithm are entered into the semester. This process is repeated until the total number of credit hours reaches a maximum. This process continues to compile for the next semester. Fig. 6 depicts the flow of the knapsack problem where the process is repeated until the number of hours and courses for each semester reaches the maximum rate.

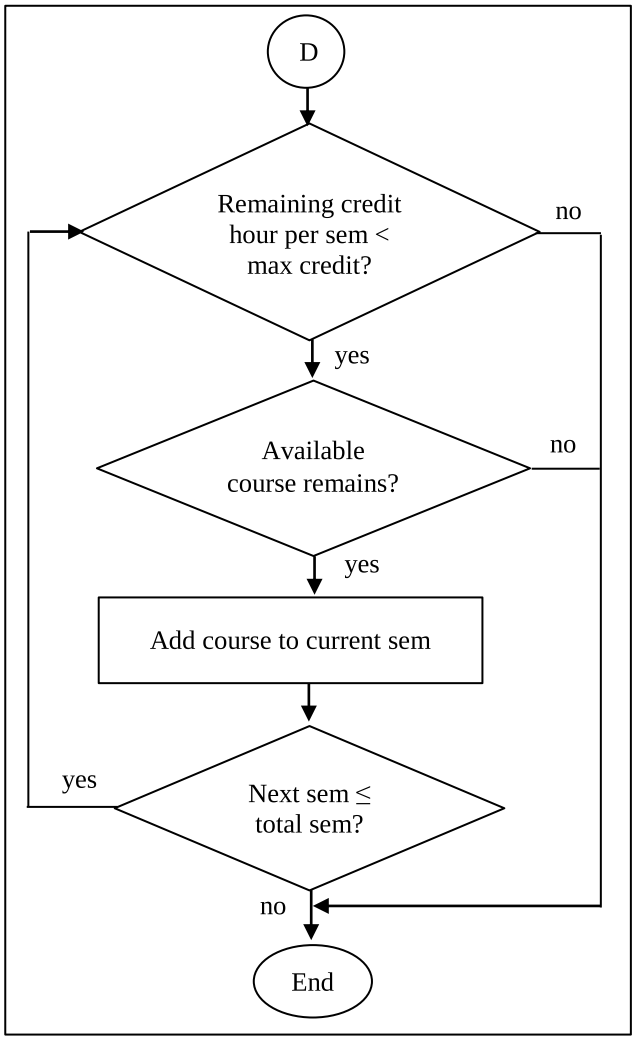
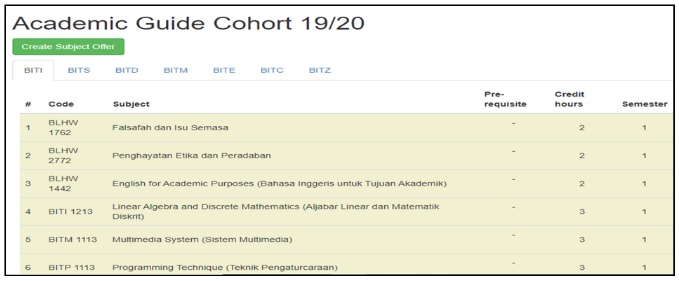

D
Remaining credit hour per sem < max credit?
no
yes
Available course remains?
no
yes
Add course to current sem
yes
Next sem ≤ total sem?
no
End

Fig. 6. Sorting Courses using Knapsack Problem 0-1.

IV. Automated Study Plan Generator Prototype

The creation of the courses details for a complete curriculum structure is the first stage in constructing this prototype. The course details include course code, course name, prerequisites course, credit hours, and semester of offering are the required input as shown in Fig. 7.

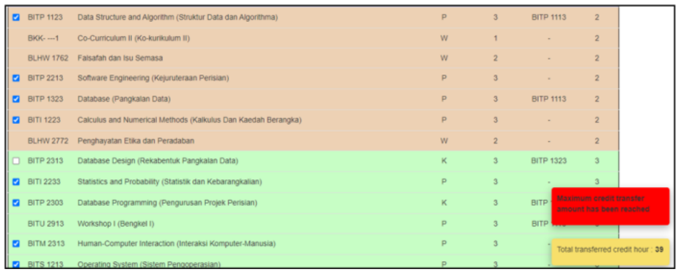


Academic Guide Cohort 19/20

#	Code	Subject	Pre-requisite	Credit hours	Semester
1	BLHW 1762	Falsafah dan Isu Semasa	-	2	1
2	BLHW 2772	Penghayatan Etika dan Peradaban	-	2	1
3	BLHW 1442	English for Academic Purposes (Bahasa Inggeris untuk Tujuan Akademik)	-	2	1
4	BITI 1213	Linear Algebra and Discrete Mathematics (Aljabar Linear dan Matematik Diskrit)	-	3	1
5	BITM 1113	Multimedia System (Sistem Multimedia)	-	3	1
6	BITP 1113	Programming Technique (Teknik Pengaturcaraan)	-	3	1

Fig. 7. Course Details.

The total credit hours exempted is then determined as shown in Fig. 8. This stage is crucial since it serves as the prototype's major support structure. The total credit hours exempted must not exceed the maximum credits set by the university. The prototype shows warning when total credit hours exempted exceed the maximum credits to prevent students from making mistakes.



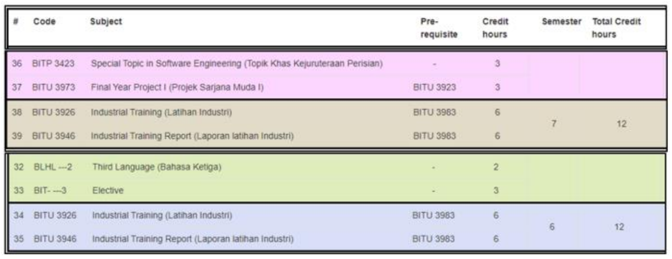
☑	BITP 1123	Data Structure and Algorithm (Struktur Data dan Algorithma)	P	3	BITP 1113	2
	BKK- ---1	Co-Curriculum II (Ko-kurikulum II)	W	1	-	2
	BLHW 1762	Falsafah dan Isu Semasa	W	2	-	2
☑	BITP 2213	Software Engineering (Kejuruteraan Perisian)	P	3	-	2
☑	BITP 1323	Database (Pangkalan Data)	P	3	BITP 1113	2
☑	BITI 1223	Calculus and Numerical Methods (Kalkulus Dan Kaedah Berangka)	P	3	-	2
	BLHW 2772	Penghayatan Etika dan Peradaban	W	2	-	2
☐	BITP 2313	Database Design (Rekabentuk Pangkalan Data)	K	3	BITP 1323	3
☑	BITI 2233	Statistics and Probability (Statistik dan Kebarangkalian)	P	3	-	3
☑	BITP 2303	Database Programming (Pengurusan Projek Perisian)	K	3	BITP [illegible]	[illegible]
	BITU 2913	Workshop I (Bengkel I)	P	3	BITP [illegible]	[illegible]
☑	BITM 2313	Human-Computer Interaction (Interaksi Komputer-Manusia)	P	3	-	[illegible]
☑	BITS 1213	Operating System (Sistem Pengoperasian)	P	3	-	[illegible]

Maximum credit transfer amount has been reached

Total transferred credit hour : **39**

Fig. 8. Exempted Courses and Credit Hours.

Subsequently, the algorithm will determine the number of semesters and arrange the courses that are appropriate for the student. According to the parameters given by the algorithm, new courses will be substituted for the exempted courses. Fig. 9 shows the example of courses plan generated from the automated study plan generator.



#	Code	Subject	Pre-requisite	Credit hours	Semester	Total Credit hours
36	BITP 3423	Special Topic in Software Engineering (Topik Khas Kejuruteraan Perisian)	-	3		
37	BITU 3973	Final Year Project I (Projek Sarjana Muda I)	BITU 3923	3		
38	BITU 3926	Industrial Training (Latihan Industri)	BITU 3983	6	7	12
39	BITU 3946	Industrial Training Report (Laporan latihan Industri)	BITU 3983	6		
32	BLHL ---2	Third Language (Bahasa Ketiga)	-	2		
33	BIT- ---3	Elective	-	3		
34	BITU 3926	Industrial Training (Latihan Industri)	BITU 3983	6	6	12
35	BITU 3946	Industrial Training Report (Laporan latihan Industri)	BITU 3983	6		

Fig. 9. Example of Courses Plan.

V. Methods

A. Expert Reviews

Expert review was conducted to evaluate the suitability and accuracy of the proposed algorithm. For this study, experts consist of lecturers who had experience as academic advisors in a faculty. In total, four experts representing various academic program were participated in the review. The experts were contacted in advance to obtain information about their experience as academic advisors and to obtain their consent to become experts. Each expert was chosen from different departments to ensure that the rules established for each program were followed correctly. Then, the test case was sent via email.

The preparation of test cases was planned following the program to be given to experienced academic advisors. Test cases were organized based on the study plan generated from the proposed algorithm. To ensure accuracy, respondents were allowed to test the automated study plan generator prototype at random. The test cases were divided into 3 sections. Section A contains five test cases of total credit exemption between 3 and 15 credit hours which allows students to take 7 semesters of study. Section B contains five test cases of total credit exemption between 18 and 36 credit hours which allows students to take 6 semesters of study. Section C contains 3 test cases based on the random credit exemption course selected by the respondents. The total credit exemption between 3 and 36 credit hours. In total, 32 test cases were distributed to the experts.

B. Testing

Testing was conducted to identify bugs in the proposed algorithm. Testing helps in understanding and refining the given requirements [20]. It is the practice of comparing a piece of software's behavior to the predetermined and expected behavior established during the development phase. This method of testing accuracy was accomplished through the use of a study plan generated by the prototype. This test was run for each program several times to identify any problems that may have arisen. This test was performed independently to ensure that the study plans produced met the study's objectives.

C. Questionnaire Survey

A user acceptance survey was developed with Google Forms and sent through messages to respondents. The survey was distributed to direct entry students who are aware of the direct entry concept and procedures. The questionnaire items were separated into sub-categories to acquire a clear understanding and accountability of evaluations and comments at the next step. The technology acceptance model (TAM) created by Davis was used in this study to evaluate the behavior of persons by using one generally known theory on the actual use behavior of utilizing new technology [21]. The influences on the intention of using the prototype were based on the individual's perceived ease of use (EU). The capability of the prototype (CP) was determined in terms of features and results generated to leverage user needs for study plan activities. Attitude (ATT) was thought to assist in meeting the needs of the users and hence influenced the attitude created toward the prototype. The perceived usefulness (PU) ensured that the developed prototype received a response in terms of use and usage behavior. Lastly, the student intends to use (IU) was created to determine the extent to which prototype requirements were developed to address existing problems. Fig. 10 illustrates the revised TAM model that specifically explains the computer acceptance determinants that are general and capable of explaining user behavior toward the automated study plan generator.

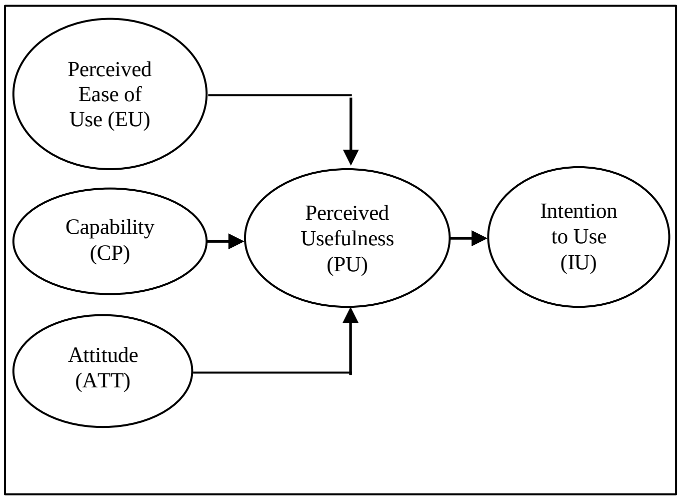


Fig. 10. Revised TAM for Automated Study Plan Generator.

The questionnaire used a five-point Likert scale of 1 to 5, 1 refers to strongly disagree and 5 refers to strongly agree. The questionnaire consists of 19 items. Respondents were instructed to use the automated study plan generator first to provide an overview of the prototype. Next, the respondents were required to answer the questionnaire survey. Each aspect presented was analyzed to gain the respondents' acceptance of the prototype to achieve the objectives.

D. Data Analysis

Test case results from experts and testing were analyzed using a confusion matrix to evaluate the accuracy. Table III shows the aspects to calculate the accuracy of the matrix by taking the average values across the "main diagonal".

The formula to calculate the accuracy based on the confusion matrix is shown in (3).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$

TABLE III. Confusion Matrix for Binary Classification

		True Class	
		Positive	**Negative**
Predicted Class	**Positive**	**TP**: Expected outcome was YES, and the actual outcome was also YES.	**FP:** The expected outcome was YES, and the actual outcome was NO.
	Negative	**FN:** The expected outcome was NO, and the actual outcome was YES.	**TN**: The expected outcome was NO, and the actual outcome was also NO.

On the other hand, a descriptive analysis (Mean ± SD) and correlation analysis were performed to analyze the data collected form the questionnaire survey.

VI. Results

Table IV shows the results from the expert review. In total, four experts representing various program were participated in the review. Each expert evaluated thirteen cases of a particular program. The results indicate that the automated study plan generator generates highly accurate study plan between 0.99 to 1 accuracy.

According to an expert who evaluated BITI program, the P1 (Workshop 1) course should be scheduled in the second semester so that students can learn how to tackle their studies at UTeM first. If a student successfully exempted for more than 18 credit hours, the algorithm rules place P1 course in Semester 1. They can be changed because the P1 course is available in both, and a new course will be taken. As for the BITC program, the arrangement of courses developed in the automated study plan generator is good, except that there is a problem with the K1 (Workshop II) course that is supposed to be taken before the end of Year 2. This is because the outcome of the arrangement produced depends on the P6 (Data Communication and Networking) course whether it is excluded or not. This plays an important role in compiling the study plan, but it is not stated in the handbook. Human error occurs where the availability of elective courses is incorrectly set causing the accuracy of the program to decrease.

Overall, the study plan generated from the proposed algorithm has high accuracy. It is very useful for new direct entry students to obtain an initial overview of the preparation of study plans at the beginning of the semester. The courses offered also depend on the quota set by the faculty, which forces students to change their study plans in the event of a change.

A. Testing

Table V shows 28 manual testing results from various programs. Random course selection reveals that the automated study plan generator prototype has an accuracy of 0.999 on an average. The accuracy of the manually tested program has given a value of 1 except for the BITE program. This is because when the BITE program is shortened; students must merge three even semesters into two semesters. This causes the generated study plan exceeds the total credit hours. If a student is exempted from 18 credit hours, but the courses provided in the second semester are not reduced, the generated study plan will be unbalanced credit hours.

B. Questionnaire Survey

Forty-four direct entry students have participated in the survey. These students were from Semesters 2 and 6. The female and male respondents were 43.2% and 56.8% respectively. BITS program had the highest percentage of 45.5%, followed by BITI and BITD at 15.9%. Besides, there are 11.4% students from the BITC program and 9.1% from the BITM program. Lastly, there are 2.3% of students from the BITZ program.

TABLE IV. Results from Expert Review

Program	TP	TN	FP	FN	Total	Accuracy
BITI	539	0	7	0	546	0.99
BITM	546	0	0	0	546	1.00
BITC	538	0	8	0	546	0.99
BITS	544	0	2	0	546	0.99

TABLE V. Testing Results

Program	TP	TN	FP	FN	Total	Accuracy
BITI	252	0	0	0	252	1
BITS	126	0	0	0	126	1
BITM	210	0	0	0	210	1
BITC	210	0	0	0	210	1
BITZ	168	0	0	0	168	1
BITE	125	0	1	0	126	0.992
BITD	84	0	0	0	84	1
Total	**1175**	**0**	**1**	**0**	**1176**	**0.999**

TABLE VI. Descriptive Analysis of User Acceptance Constructs

Construct	Mean ± SD
Perceived ease of use (EU)	4.301 ± 0.610
Perceived usefulness (PU)	4.291 ± 0.624
Capability (CP)	4.369 ± 0.561
Attitude (ATT)	4.348 ± 0.618
Intention to use (IU)	4.242 ± 0.619

Table VI shows a descriptive analysis of the acceptance test constructs. All mean values are greater than 4.2, indicating that respondents have a generally positive opinion of the automated study plan generator. A total of 96% of respondents agreed that the automated study plan generator is capable of producing a study plan that meets the specified requirements. The majority of respondents (92%) rated all items under attitude and perceived ease of use constructs on a scale of 4 (agree) to 5 (strongly agree). All respondents also agreed on the automated study plan generator's perceived usefulness. Lastly, the automated study plan generator would be used by more than 90% of the respondents.

Correlation analysis of the acceptance test between constructs is shown in Table VII. The results indicate all the constructs show a positive and strong correlation (exceeding 0.5), with all correlations significant at the $p<0.01$ level. The relationship between capability (CP) and intention of use (IU) is 0.862, indicating that the two are highly correlated. The finding implies that user intention is based on the capabilities of the prototype to assist users in achieving the goal of use.

TABLE VII. Correlation Analysis of User Acceptance Constructs

Construct	EU	PU	CP	ATT	IU
EU	1				
PU	0.842	1			
CP	0.671	0.743	1		
ATT	0.638	0.774	0.769	1	
IU	0.600	0.753	0.862	0.801	1

Following that is perceived ease of use (EU) concerning perceived use (PU), with a high correlation between the two constructs, demonstrating that the prototype is simple to understand and provides convenience to the user.

VII. Discussion

An algorithm for compiling study plans was proposed and validated in this study. Rule-based and knapsack problem were applied in compiling student learning plans. The rule-based method is utilized to optimize the courses that students must take during the semester as specified by the faculty. These courses have been planned based on the total number of credit hours exempted. There are crucial courses that must be prioritized based on the semester to guarantee that students do not miss out and create a change in the intended number of semesters. Besides, the knapsack problem used in this study is intended to select courses that are not included in the rules and can be put into a table based on credit hours and the desired offer. The courses to be chosen are balanced according to the number of hours allotted. As a result, a study plan that satisfies the prerequisites is created. These two approaches are ideal for dealing with this issue. This is because significant courses can be certain of their offer, while other courses are offered following the correct offer. The planned structure qualifies for making a study plan. The results from expert reviews and testing reveal that the automated study plan generator prototype has an accuracy of 0.999 on an average. Moreover, most of the respondents participated in the user acceptance survey have a generally positive opinion of the automated study plan generator in term of ease of use, usefulness and capability. The automated study plan generator would be used by more than 90% of the respondents.

The results produced from this study could provide valuable contributions to the undergraduate students to plan their course schedule prior to their graduation. The process of organizing learning can be more effectively implemented using the proposed algorithm. Students will not be overburdened and will be able to increase the consistency of their learning output in the coming semester by finding suitable learning arrangements. It has the potential to indirectly improve student learning performance.

This study has certain limits and problems. If the rule is incorrect or not written in the academic handbook, it can disrupt the schedule's arrangement. This is because the rules cannot be followed, resulting in a wild and incorrect arrangement. It will stymie students' planning and make new arrangements difficult. When constructing the study plan, Knapsack Problem 0-1 acts greedily to avoid this problem by using rules to ensure the algorithm obeys the established limitations. Knapsack Problem 0-1 will continue to produce results based on the number of credit hours without following the course codes. Moreover, human error is unavoidable when conducting studies, which reduces the accuracy of the results. The combination ruled-base and knapsack problem algorithm assures that the study schedule can be organized properly. Prerequisites can be met, credit hours can be allocated in a balanced manner, and courses can be arranged according to the offers by the faculty, all in one system. With the study's findings, any desired method can be constructed in the future. To solve this problem, rule-based and knapsack problem are appropriate. This is because each course must follow all rules, and voids can be filled by the knapsack problem with greedily picked courses to fulfil the prerequisite credit hours. It also relies on the availability of courses offered in the semester. The results have a high level of accuracy and can be used by academic advisors and new direct entry students to arrange their schedules.

VIII. Conclusion

This study had successfully proposed and validated an algorithm for compiling study plans by using rule-based and knapsack problem. Based on the results and analysis, it is possible to conclude that the accuracy of the algorithm based on the rule-based and knapsack problem to generate study plan is high. Survey respondents believe that the proposed algorithm can assist them in creating and designing study plans. The majority of respondents are interested in using the automated study plan generator. Finally, it can be seen that both students and academic advisors can benefit from the automated study plan generator to arrange their study plans.

Acknowledgment

This study is supported by the Universiti Teknikal Malaysia Melaka research grant (JURNAL/2020/FTMK/Q00055).

References

[1] T. R. Hill and S. D. Roberts, "A prototype knowledge-based simulation support system," Simulation, vol. 48, no. 4, pp. 152–161, 1987, doi: 10.1177/003754978704800407.

[2] H. Yang, M. Anbarasan, and T. Vadivel, "Knowledge-Based Recommender System Using Artificial Intelligence for Smart Education," J. Interconnect. Networks, vol. 2143031, 2022.

[3] M. Á. Rodríguez-García, F. García-Sánchez, and R. Valencia-García, "Knowledge-Based System for Crop Pests and Diseases Recognition," Electronics, vol. 10, no. 8, p. 905, 2021.

[4] M. R. Khosravani, S. Nasiri, and T. Reinicke, "Intelligent knowledge-based system to improve injection molding process," J. Ind. Inf. Integr., vol. 25, no. August 2021, p. 100275, 2022, doi: 10.1016/j.jii.2021.100275.

[5] S. Bashir, A. A. Almazroi, S. Ashfaq, A. A. Almazroi, and F. H. Khan, "A Knowledge-Based Clinical Decision Support System Utilizing an Intelligent Ensemble Voting Scheme for Improved Cardiovascular Disease Prediction," IEEE Access, vol. 9, pp. 130805–130822, 2021, doi: 10.1109/ACCESS.2021.3110604.

[6] W. Y. Zhang, S. B. Tor, and G. A. Britton, "A prototype knowledge-based system for conceptual synthesis of the design process," Int. J. Adv. Manuf. Technol., vol. 17, no. 8, pp. 549–557, 2001, doi: 10.1007/s001700170137.

[7] F. Mustapha, N. Ismail, S. M. Sapuan, Z. Noh, and A. Samsuri, "Development of a prototype knowledge-based system for troubleshooting of aircraft engine and parts - A case study of Cessna Caravan," Int. J. Mech. Mater. Eng., vol. 5, no. 1, pp. 36–42, 2010.

[8] I. H. Sarker, "AI-Based Modeling: Techniques, Applications and Research Issues Towards Automation, Intelligent and Smart Systems," SN Comput. Sci., vol. 3, no. 2, pp. 1–20, 2022, doi: 10.1007/s42979-022-01043-x.

[9] G. Engin et al., "Rule-based expert systems for supporting university students," Procedia Comput. Sci., vol. 31, pp. 22–31, 2014, doi: 10.1016/j.procs.2014.05.241.

[10] I. H. Sarker, A. Colman, J. Han, and P. Watters, "Context-Aware Rule-Based Expert System Modeling BT - Context-Aware Machine Learning and Mobile Data Analytics: Automated Rule-based Services with Intelligent Decision-Making," I. Sarker, A. Colman, J. Han, and P.

Watters, Eds. Cham: Springer International Publishing, 2021, pp. 129–136.

[11] K. Qu, T. Gong, and J. Shao, "Design and implementation of system generator based on rule engine," Procedia Comput. Sci., vol. 166, pp. 517–522, 2020, doi: 10.1016/j.procs.2020.02.054.

[12] S. S. Raghavan, "Rule Based Heuristic Approach for Minimizing Total Flow Time in Permutation Flow Shop Scheduling," Teh. Vjesn., vol. 22, no. 1, pp. 25–32, 2015, doi: 10.17559/TV-20130704132725.

[13] D. Sapra, R. Sharma, and A. P. Agarwal, "Comparative study of metaheuristic algorithms using Knapsack Problem," Proc. 7th Int. Conf. Conflu. 2017 Cloud Comput. Data Sci. Eng., pp. 134–137, 2017, doi: 10.1109/CONFLUENCE.2017.7943137.

[14] J. Lv, X. Wang, M. Huang, H. Cheng, and F. Li, "Solving 0-1 knapsack problem by greedy degree and expectation efficiency," Appl. Soft Comput. J., vol. 41, pp. 94–103, 2016, doi: 10.1016/j.asoc.2015.11.045.

[15] M. D. Goodman, K. A. Dowsland, and J. M. Thompson, "A grasp-knapsack hybrid for a nurse-scheduling problem," J. Heuristics, vol. 15, no. 4, pp. 351–379, 2009.

[16] R. O. K. Base and P. Nilaphruek, "Automatic Course Planning System Using Rule-Based Ontological Knowledge Base," Int. J. Comput. Internet Manag. Vol.23, vol. 23, no. 1, pp. 16–23, 2015.

[17] N. Chanamarn and K. Tamee, "Enhancing Efficient Study Plan for Student with Machine Learning Techniques," Int. J. Mod. Educ. Comput. Sci., vol. 9, no. 3, pp. 1–9, 2017, doi: 10.5815/ijmecs.2017.03.01.

[18] T. Sobh, S. Patel, and R. Cousen, "Course Scheduler : An Automated Schedule Generator," 2007.

[19] X. Wang, Y. Bai, C. Cai, and X. Yan, "A production rule-based knowledge system for software quality evaluation," ICCET 2010 - 2010 Int. Conf. Comput. Eng. Technol. Proc., vol. 6, pp. 208–211, 2010, doi: 10.1109/ICCET.2010.5486303.

[20] C. Klammer and R. Ramler, "A Journey from Manual Testing to Automated Test Generation in an Industry Project," Proc. - 2017 IEEE Int. Conf. Softw. Qual. Reliab. Secur. Companion, QRS-C 2017, pp. 591–592, 2017, doi: 10.1109/QRS-C.2017.108.

[21] R. Rauniar, G. Rawski, J. Yang, and B. Johnson, "Technology acceptance model (TAM) and social media usage: An empirical study on Facebook," J. Enterp. Inf. Manag., vol. 27, no. 1, pp. 6–30, 2014, doi: 10.1108/JEIM-04-2012-0011.