JP3I (Jurnal Pengukuran Psikologi dan Pendidikan Indonesia)
DOI: http://dx.doi.org/10.15408/jp3i.v14i1.32091
http://journal.uinjkt.ac.id/index.php/jp3i

Development of *irtawsi*: A User-Friendly R Package for IRT Analysis

Hari Purnomo Susanto[1,2], Agus Maman Abadi[3], Haryanto[4], Heri Retnawati[1], Raden Muhammad Ali[5], Hasan Djidu[6]

Educational Research and Evaluation, Graduated School, Universitas Negeri Yogyakarta, Indonesia[1]
STKIP PGRI Pacitan, Indonesia[2]
Faculty of Mathematics and Natural Science, Universitas Negeri Yogyakarta, Indonesia[3]
Faculty of Engineering, Universitas Negeri Yogyakarta, Indonesia[4]
Faculty of Teacher Training & Education, Universitas Ahmad Dahlan, Indonesia[5]
Faculty of Teacher Training and Education, Universitas Sembilanbelas November Kolaka, Indonesia[6]

Email: haripurnomo.2021@student.uny.ac.id, haripurnomosusanto@gmail.com

Abstract

The complexity of the IRT analysis makes it difficult to perform manually, therefore requiring easy-to-use software. While many software options exist for IRT analysis, the high cost of paid software can make it inaccessible for many students and lecturers in Indonesia. While the *mirt* package provides a complete, free option for IRT analysis, proficiency in the R programming language is required. This study aims to develop an R package for IRT analysis, equipped with a user-friendly interface based on the *mirt* package, designed to be easy to use for beginners in IRT analysis. The System Development Life Cycle (SDLC) model is used for development and includes five stages: Planning, Analysis, Design, Implementation, and System. The resulting package is named *irtawsi* and includes functionality comparable to paid software. This package can calibrate both test and non-test instruments using various IRT models, such as the Rasch, 2PL, 3PL, 4PL, GRM, PCM, and GPCM models. The *irtawsi* package functionality includes: (1) an easy-to-use user interface, (2) automatic interpretation of analysis results, (3) a guide for IRT analysis, (4) recommendations when assumptions are not met, (5) an HTML report format for analysis results,(6) support for two languages (Indonesian and English), (7) it is free, and (8) can be installed on Windows, macOS, and Linux operating systems. The results of this development contribute to the calibration process, making it easier for practitioners and researchers to calibrate the instruments being developed, especially for beginners who are learning IRT.

Keywords: IRT, irtawsi, instrument, calibration, user friendly

Abstrak

Kompleksitas dari IRT membuatnya susah digunakan secara manual, sehingg membutuhkan software yang mudah digunakan. Banyak software yang dapat digunakan untuk analisisi IRT. Software berbayar memiliki fitur yang mudah digunakan pada mode full versi, Namun harga yang mahal banyak membuat mahasiswa dan dosen di Indonesia tidak bisa menggunakannya. Paket mirt merupakan salah satu paket gratis yang lengkap untuk analisis IRT, namun pengguna harus menguasai bahasa pemograman R untuk menggunakannya. Tujuan dari studi ini yaitu mengembangkan paket analisis IRT yang dilengkapi dengan user interface berbasisi paket mirt yang mudah digunakan untuk pemula yang sedang belajar IRT. Model pengembangan yang digunakan yaitu System Development Life Cycle (SCLD). Terdapat lima tahapan dalam model ini dianranya, Planning, Analysist, Design, Implementation, dan System. Paket yang dikembangkan diberi nama irtawsi.Pengembangan ini menghasilkan paket irtawsi dengan fungsionalitas yang tidak kalah dengan software-software berbayar. Paket ini mampu

melakukan kalibarasi instrumen tes dan non tes. Model IRT yang dapat digunakan pada paket ini yaitu model Rasch, 2PL, 3PL, 4PL, GRM, PCM dan GPCM. Fungsionalitas paket irtawsi diantaranya yaitu (1) mudah digunakan dengan user interface, (2) mampu memberikan interpretasi hasil analisis secara otomatis, (3) dilengkapi dengan panduan atau Langkah-langkah analisis IRT, (4) mampu memberikan saran ketika uji asumsi tidak terpenuhi, (5) Hasil analisis dapat diunduh dalam bentuk dokumen laporan dengan ekstensi html, (6) dapat menggunakan dua bahasa yaitu bahasa Indonesia dan Inggris, (7) Gratis, dan (8) compatible dengan sitem operasi Windows, macOS, dan Linux. Hasil pengembangan ini memiliki kontribusi dalam proses kalibrasi yang memberikan kemudahan bagi praktisi dan peneiliti dalam kalibrasi instrumen yang sedang dikembangkan, terutama bagi para pemula yang sedang belajar IRT.

***Kata kunci**: IRT, irtawsi, instrumen, kalibrasi, user friendly*

Introduction

Item Response Theory (IRT) has many advantages over Classical Test Theory (CTT). However, IRT has a more complex structure than CTT (Bichi & Talib, 2018; De Champlain, 2010), so IRT may not be widely used without software.

Many IRT software programs can be used for item calibration, such as *IRTPRO 2.1*, *MPLUS* 7.1, Flex*MIRT*, *EQSIRT* (Han & Paek, 2014), *BILOG-MG*, and *PRSCALE* (Choi & Asilkalkan, 2019). These software programs are easy to use to determine item and person characteristics. However, these programs can only be used for free in the trial mode and require high fees for full access (Choi & Asilkalkan, 2019; Han & Paek, 2014). Therefore, IRT-based calibration is not accessible to some instrument developers (Foster et al., 2017), and the high cost of IRT software may not be affordable for most teachers, lecturers, and students in Indonesia.

One of the software programs that can be used for free for IRT analysis is the *R Program* (R Core Team, 2022), which has many packages for IRT analysis. One of the packages for a complete IRT analysis is the *mirt* package (Chalmers, 2012; Choi & Asilkalkan, 2019; Hori et al., 2020), which can analyze unidimensional and multidimensional IRT models. Instructions for using this package can be found in the Comprehensive R Archive Network (CRAN), but users must understand command codes in the R programming language.

The *R program* can only be used by users familiar with data analysis methods and the R programming language (Hackenberger, 2020; Paura & Arhipova, 2012). Similarly, IRT users who want to use *mirt* packages (Chalmers, 2012) must master command codes in the R programming language. This package has powerful capabilities for IRT analysis, but it is not easy to use. For beginners in the IRT, using the *R program* and the *mirt* package is not easy.

Several packages for IRT analysis based on the *R program* have been developed. These packages include *irtGUI* (Yildiz, 2021), *IRTshiny* (Hamilton & Mizumoto, 2017), and *catIRTtools* (Aybek, 2021). Users can use them without mastery of the R programming language. However, the features do not support beginners who are still learning IRT.

IRT practitioners greatly benefit from the previously mentioned IRT software, especially the free version with a user interface. In order to become competent with software, IRT experts who are not yet accustomed to it require only a minor adjustment. Nevertheless, this software requires significant time for novice IRT practitioners to become proficient. Beginners must comprehend IRT concepts in addition to conquering the software. Novices will probably face challenges and even make errors in their comprehension of IRT during this process.

Like most statistical concepts. The IRT concept also involves assumptions that must be met. (Balafas, Spyros et al., 2020; Bichi & Talib, 2018; DeMars, 2010; Hambleton et al., 1991; Reise et al., 2021; Sijtsma

& van der Ark, 2022). Violations of these assumptions may lead to invalid result interpretations (Foster et al., 2017; Hu & Plonsky, 2021; Shatz, 2023). Foster et al. (2017) discovered that only 6.78% of traditional IRT articles included complete assumption testing, and 33.9% did not mention it. The study did not provide an explicit explanation for the cause. The practitioners' lack of knowledge may have influenced this case. Practitioners were either unable to conduct any assumption testing or were unable to satisfy them, which left them uncertain about how to proceed.

In addition to assumption testing, other challenges are associated with applying IRT. First, practitioners or users cannot interpret the results of the analysis (Foster et al., 2017). This issue also arises with simpler statistical analyses like regression (Shatz, 2023). Secondly, Foster et al. 2017 noted a deficiency in comprehensive instructions or illustrations for implementing IRT. These guidelines can be in the form of concepts containing steps in IRT analysis or instructions for using analysis software. Most software with a user interface has a separate guide from the software itself. Third, it is important to understand the language used in the software (Cavaliere, 2015). If users or practitioners do not understand the software's explanations, their problems will worsen. Fourth, there is a need for more interactive and user-friendly IRT software (Foster et al., 2017).

IRT software must be interactive and simple to use. The *irtGUI, IRTshiny*, and *CatIRTtolls* packages have met this criterion. However, it would be beneficial if the software also included informative features. These features include the ability for the software to detect errors from the beginning of the analysis and notify the user. Secondly, the software integrates the user guide, eliminating the need for users to search for it externally.

Additionally, the guidelines include information about the procedures used to conduct IRT analysis. Thirdly, the software undergoes analysis using the IRT concept. The fourth feature involves elucidating the analysis results to minimize errors in IRT interpretation. If the system fails to meet any assumptions, it can offer recommendations. Finally, selecting the right language helps users navigate and comprehend the software.

Based on the description above, a program for easy and free IRT analysis is urgently needed. To realize the existence of an application that can be used for IRT analysis for free, this study aims to develop an R package for IRT analysis so that it can be used without mastery of the R programming language. The developed package has features: (1) easy-to-use interactive user interface, (2) interpretation of analysis results, (3) equipped with a guide or IRT analysis steps, (4) advice when the assumption test is not met, (5) easy to download analysis report, (6) available in Indonesian and English, and (7) free of charge. These features are expected to be a tool for IRT analysis and a learning medium for beginners in IRT.

Methods

Development Model

The model used in developing this package is the System Development Life Cycle (SDLC) (Dennis et al., 2015; Tilley & Rosenblatt, 2016). The stages of SCLD are very strict because completing a stage must be carefully calculated before proceeding to the next stage. This can minimize errors in application development. These characteristics make SDLC very suitable for developing statistical applications. Stages in the SDLC include: (1) Planning: determining development goals, priority programming languages to be used, and product names. (2) requirements analysis consists of (a) determining the IRT models that can be analyzed, (b) adjusting the IRT Analysis Steps with the user interface layers of the package, and (c) determining R packages to build the user interface, IRT analysis calculations, build analysis report documents, and retrieve data. (3) Design: focus on forming user interface connectivity designs, servers, documents resulting from analysis, and user interface layer designs. (4) Implementation: coding, testing, and publishing the package to databases such as CRAN and GitHub. (5) System: describes the package's compatibility with PC operating systems, maintenance, and package updates.

Content Material

IRT analysis forms the basis of this development. The development focuses on the unidimensional IRT model, which includes dichotomous and polytomous responses. The model with dichotomous responses consists of the one-parameter logistic model (1PL), two-parameter logistic model (2PL), three-parameter logistic model (3PL) (DeMars, 2010; Retnawati, 2015), and four-parameter logistic model (4PL) (Chalmers, 2012; Paek & Cole, 2019). Furthermore, the models with polytomous responses used are the Graded Response Model (GRM), Partial Credit Model (PCM), and Generalized Partial Credit Model (GPCM) (Chalmers, 2012; Paek & Cole, 2019).

Analysis of item instruments or calibration of unidimensional IRT-based instruments using the models above can be carried out in four steps, namely (1) Determine the Fit Model, (2) Test assumptions based on the Fit Model. (3) Determine item fit and estimate item parameters, and (4) Determine the information function of the test (Retnawati, 2014). Each step is explained in detail as follows.

Determining the FIT Model

Statistical methods can be used to determine model fit, including: the likelihood ratio test, M2 and C2 (Chalmers, 2012; Paek & Cole, 2019). The likelihood ratio test method is called relative fit (Auné et al., 2020). This method utilizes the coefficient of the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), Sample-Size Adjusted BIC (SABIC), and Hannan-Quinn (HQ). These coefficients can be used if at least two models are being compared. This coefficient does not mean anything if only one IRT model is used. The model that has the smallest coefficient value is said to be the most fit model. Furthermore, the M2 method (Maydeu-Olivares, 2013, 2014) and C2 (Cai & Monroe, 2014) can be referred to as the global fit (Auné et al., 2020), which can be used to determine model fit. The fit criteria for both models are met if the P-value M2 is >0.01, RMSE <0.8, and CFI >0.9 (Maydeu-Olivares, 2014).

IRT Assumption

As in most parametric statistics, item analysis with IRT must meet several assumption tests, namely (1) unidimensional, (2) local independence, and (3) parameter invariance (Hambleton et al., 1991; Jumailiyah, 2017; Reeve et al., 2007; Retnawati, 2014).

a. Unidimensional Assumption

This assumption can be proved by the eigenvalues in the invariance matrix and inter-item covariance resulting from factor analysis (Retnawati, 2014). Proof with this method can be done in two ways. First, the percentage of variability the dominant factor explains is more than 20% (Lameijer et al., 2020; Retnawati, 2014). Second, using the Scree plot. Visually, this assumption is fulfilled if there is only one steep slope (Retnawati, 2014). These methods can be applied if the minimum sample size is met (Retnawati, 2014).

Instruments that meet this assumption show that the instrument measures one latent trait (Retnawati, 2014) or only one construct (Lameijer et al., 2020). Conversely, the instrument measures more than one construct if it is not met. This case can be overcome by multidimensional IRT analysis. Proof of this assumption can be done using the psych package (Battauz, 2020).

b. Local Independence Assumption

This assumption explains that: (1) student responses to one item are not influenced by their responses to other items (Jumailiyah, 2017; Sudaryono, 2013); (2) one instrument item does not have a significant relationship with the other items (Reeve et al., 2007; Sudaryono, 2013). Chalmers (Chalmers, 2012) mentions five methods that can be used to prove this assumption, namely the LD method (Chen & Thissen, 1997), the Q3 method (Yen, 1984), the JSI method (Edwards et al., 2018), the *exp*, and the *expfull*

method (Chalmers, 2012). The methods used in the development are the Q3 and LD methods. This assumption is fulfilled if the absolute value of each element of the Q3 matrix <0.2236 or the absolute value of the Cramer V (LD) matrix <0.174 (Paek & Cole, 2019).

There are several ways to overcome when this assumption is not met, namely (1) delete one item from each pair of items that causes this assumption not to be met and Recalibrate (Toland, 2014) or (2) Use the Non-Parametric IRT model (Petersen, 2005); or (3) If there are two or more items that cause local dependencies and conceptually have no content related to each other (Nguyen et al., 2014) or the same procedure, then the test results can be ignored; or (4) Ignore the results of this Local Independence Assumption Test. Following the opinion of (Hambleton et al., 1991; Retnawati, 2016), if the Unidimensional Assumption is met, the local independence assumption is automatically met.

c. Parameter Invariant Assumption

Parameter invariance is an important assumption in Item Response Theory (IRT), as it ensures that the test scores are comparable across different groups of test takers. The first assumption of parameter invariance states that the item parameters are the same across different groups of test takers. To test this assumption, the participants are split into odd and even samples, and the item parameters are estimated separately for each sample using the same IRT model. The item parameters from the two samples are then correlated, and if there is a significant correlation coefficient, the first assumption of parameter invariance is met (Retnawati, 2014).

The second assumption of parameter invariance states that the ability parameters are the same across different test items (Retnawati, 2014). To test this assumption, the test items are split into odd and even sets, and the item parameters are estimated separately for each set using the same IRT model in the first step. The ability parameters are then estimated by using the item parameters, and a pair of abilities is obtained for each participant. The ability parameters from the two sets of items are then correlated, and if there is a significant correlation coefficient, the second assumption of parameter invariance is met.

If the item parameters cause either of these assumptions not to be met, then the non-invariant items should be deleted from the test. This can be done by consulting with experts or using statistical methods to identify the items that cause non-invariance (Guenole & Brown, 2014; Xu et al., 2020). By ensuring parameter invariance, the test scores can be compared across different groups of test takers, essential for making fair and accurate decisions based on the test results.

d. Increasing monotonicity

Monotonicity in IRT is a phenomenon where the probability of a person supporting or answering an item correctly increases with the improvement of their ability level (Nguyen et al., 2014; Reise et al., 2013). In the model with dichotomous responses, this assumption can be determined by examining the ICC (Hambleton et al., 1991) using the method developed by Mokken. Sub (Chalmers, 2012; Paek & Cole, 2019). Sequently, in the polytomous model, it can be determined using the Mokken method (Ark, 2007, 2012).

Violation of this assumption can lead to incorrect interpretation of item parameters. If this assumption is not met, IRT analysis can be conducted using non-parametric models or Unfolding (Reise et al., 2013).

Determining item fit and parameter estimation.

This step is carried out after all IRT assumptions have been met. Item fit is determined by looking at the Chi-Square (χ^2) value or the p-value of χ^2. If the p-value has a value of χ^2 more than 0.05, the item is said to be fit or used (Paek & Cole, 2019; Retnawati, 2014).

Test Information Function (TIF)

The Test Information Function (TIF) is a method that explains the strength of the items on the instrument in uncovering the latent trait being measured (Retnawati, 2014). TIF is the sum of the item information functions (IIF) (Vander Linden & Hambleton, 1997). TIF can be used to calculate Standard Error Measurement (SEM). SEM has a value of 1 to the root of TIF (Hambleton et al., 1991). The TIF and SEM intersection results can be used to determine the appropriate range of abilities to be measured using fit test items (Retnawati, 2014).

Parameter Ability

Parameter ability is estimated by using fit items. Chalmers (2012) states that seven methods of estimating the ability parameter exist. Four methods used in the development of this package are Expected A-Posteriori (EAP), Maximum A-Back (MAP), Maximum Likelihood (ML), and Weighted Likelihood Estimation (WLE). Because there is not much time, the remaining three methods will be added in the next version upgrade. Item parameters function as scoring using IRT (Retnawati, 2014). The parameter estimates shown in the package still use the whole item. The resulting estimated score cannot be used if items do not fit. Alternatively, users can use the cat R package (Magis & Barrada, 2017) for ability parameter estimation.

Results and Discussion

This section explains how the IRT content is packaged in the user interface development using the SCLD model. The results of each stage are described as follows:

Planning

This stage produces several decisions related to development objectives, the programming language used, and the name of the package being developed. The purpose of developing the package has been described in detail, along with its features in the background section. The package developed prioritizes using the R programming language R (R Core Team, 2023) with the help of *R Studio* software (Posit team, 2023). The developed package is expected to be published on the CRAN database and Github. Furthermore, the developed package is *irtawsi* (Item Response Analysis with Steps and Interpretation).

Analysis

The results of the first analysis stage were to decide on the seven IRT models used in *irtawsi*. The seven models consist of four dichotomous models and three polytomous models. The four dichotomous models referred to are the one-parameter logistic model (1PL), two-parameter logistic model (2PL), three-parameter logistic model (3PL), and four-parameter logistic model (4PL) (Chalmers, 2012). Furthermore, the three polytomous models are the Graded Response Model (GRM), Partial Credit Model (PCM), and Generated Partial Credit Model (GPCM).

The second analysis result was to adapt the IRT analysis steps to the many layers in the package. Six user interface layers are used in the *irtawsi* package. The first IRT analysis step in the method section becomes the second step in the *irtawsi* package, and so on. These layers are presented in the form of IRT analysis steps. For more details, refer to Table 1.

Table 1. Requirements analysis for IRT analysis using *irtawsi*.

Layer	IRT analysis steps using *irtawsi*	Description
1	Step 0. Choose Language	Additional steps
2	Step 1. Input Data and Early Detection	
3	Step 2. Determining Fit Model	Analysis steps according to the reference
4	Step 3. IRT assumption	
5	Step 4. Determining Fit Item and Parameter Estimation	
6	Step 5. Determining Test Information Function (TIF)	

Sources: Personal data (2024).

The third requirement analysis results in determining the R packages for building a user interface, conducting IRT analysis calculations, documenting the analysis results, and importing data. The detailed results of this requirement analysis can be seen in Table 2.

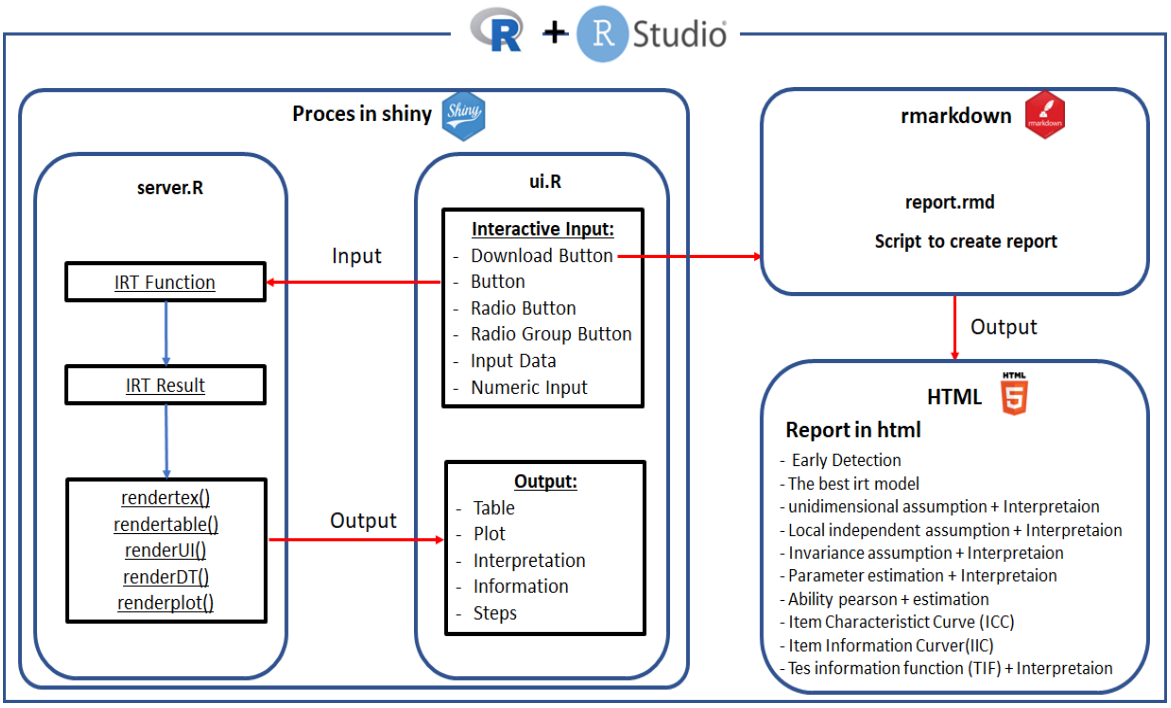
Table 2. The requirement analysis of R packages for supporting *irtawsi*

Component	R package	Package functionality
Interface	*shiny*(Perrier et al., 2023),	To build a user interface consisting of the *ui.R* and *server.R* sections
	bs4Dash(Granjon, 2022),	To upgrade the *Shiny* appearance
	*shiny*Widget(Perrier et al., 2023)	To create a button, radio button, and radio group button.
	shinycssloaderss(Sali & Attali, 2020)	To display a progress bar
	DT (Xie et al., 2023)	To display a table
	gt (Iannone et al., 2022)	To create a customized table appearance
	Diagram (Soetaert, 2020)	To create a diagram of the IRT analysis steps
IRT Calculations	*mirt* (Chalmers, 2012)	To conduct an item analysis
	Psych (William Revelle, 2023)	To determine the unidimensional assumption
	Stats (R Core Team, 2023)	To calculate the correlation in the invariance assumption analysis
Report Documentation	*rmarkdown* (Xie et al., 2020)	To create an automated report documentation of the analysis results in HTML format
Input Data	*readxl* (Wickham & Bryan, 2023)	To import data in Excel format

Sources: Personal data (2024).

Design

This step develops the connectivity component design and user interface design. The connectivity design for *Shiny* and R *markdown* components can be seen in Figure 1.



Sources: Personal data (2024).

Figure 1. The Connectivity Design of *irtawsi*

The connectivity design in Figure 1 is realized in six layers (pages). Starting from layers 2 to 6, it describes the steps of IRT analysis, resulting in a user interface design as shown in Figure 2.

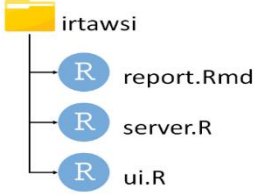
Sources: Personal data (2024).

Figure 2. User Interface Design of *irtawsi* Package

Implementation

Coding

R language is used for coding. The coding process is performed using *R Studio* software (Posit team, 2023). The coding results consist of three main files, namely (1) *ui.R* file containing user interface control codes, (2) *server.R* file containing codes from *mirt*, psych, readxl, diagram, and stats packages, and (3) report.Rmd file containing codes from rmarkdown package, which is used for report documentation file generation. These files are stored in the *irtawsi* folder and have a hierarchical structure as shown in Figure 3.



Sources: Personal data (2024).

Figure 3. The arrangement of files resulting from the *irtawsi* package coding.

Testing Process

Testing is performed by comparing the output of *irtawsi* with the output of *mirt, psych,* and *stats*. The testing process is repeated several times using data in dichotomous and polytomous formats. The testing results with experts are shown in Table 3.

Table 3. Testing Result of *irtawsi* package

Features functionality	***Irtawsi* functionality**
Early detection	√
The best IRT model	√
Unidimensional assumption	√√
Local Independencies assumption	√
Invariance assumption	√√√
Item Parameter estimation	√
Person ability estimation	√
Item Characteristic Curve (ICC)	√
Item Information Curve (IIC)	√
Test Information Function (TIF)	√

- The √ symbol indicates the same result as the manual analysis output of the *mirt* package.
- The √√ symbol indicates the same result as the manual analysis output of the psych package.
- The √√√ symbol indicates the same result as the manual analysis output of the *mirt* + stats package.

Sources: Personal data (2024).

Publishing the Irtawsi package to CRAN and GitHub

The package can be published as open source on the GitHub and CRAN databases. The package publication on CRAN is performed using *R Studio* software and this package (Wickham et al., 2022). After going through several processes, the *irtawsi* package was published on CRAN on March 26, 2023, with version 0.3.4, and on June 26, 2024, it was updated to version 0.4.1.(Susanto et al., 2024). All information related to the *irtawsi* package and its codes can be accessed at the link https://cran.r-project.org/web/packages/*irtawsi*/index.html or the link https://github.com/SusantoHP/*irtawsi*.

System

The system used for developing the *irtawsi* package is different from web-based systems. The developer does not need a database to distribute the *irtawsi* package to users. Users only need to install the package by downloading it from CRAN or GitHub. The advantage of a package being published on CRAN is that the developer does not need to create multiple package versions to be compatible with the operating systems used by users. The CRAN volunteer community determines the package compatibility for each operating system. The *irtawsi* package is compatible with Windows, Linux, and MacOS operating systems. This information can be accessed at the link https://cran.unimelb.edu.au/web/checks/check_results_*irtawsi*.html

There are two ways to update the package. The first is to repair the package if errors have not been detected during testing. Second, adding new features and functionality to the package.

Installation and Illustration

Installation of the irtawsi Package

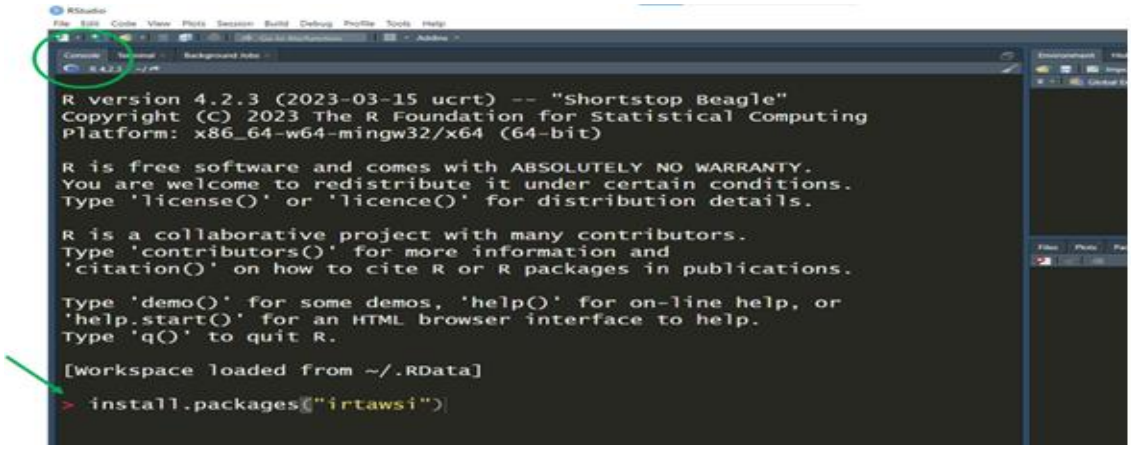
To use the *irtawsi* package, the user must install the *R program* alone or along with *R Studio*. The *irtawsi* package can be installed via CRAN by typing the following code on the console:

>install.packages("*irtawsi*")

Alternatively, it can be installed via GitHub by typing the following code on the console:

> devtools::install_github("SusantoHP/*irtawsi*")

Press enter and wait for the installation process to complete. An example of installing the *irtawsi* package through *R Studio* can be seen in Figure 4.



Sources: Personal data (2024).

Figure 4. Installing the *irtawsi* package through *R Studio*.

Illustration of the irtawsi Package

This section describes the package's features in each layer and how to use them. These features consist of interactive and informative features. The following is a simulation of IRT analysis using the *irtawsi* package.

After installing the *irtawsi*, type the following code on the console to run the package:

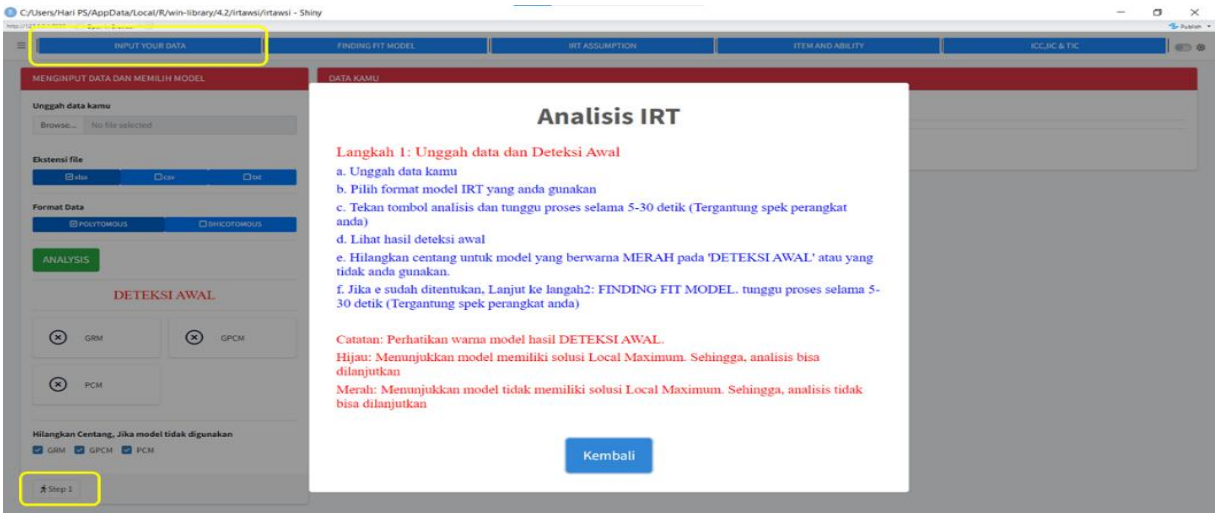
>*irtawsi*::*irtawsi*() or >library(*irtawsi*)
>*irtawsi*()

Then press enter, and a display like Figure 5 will appear.

Sources: Personal data (2024).

Figure 5. Layer 1 of the *irtawsi* package.

Figure 5 shows the display in layer one or step 0. There are three features: (1) choosing the language used, in this case, the language used is Indonesian; (2) information about the *irtawsi* package; and (3) steps for IRT analysis. Next, click the "input your data" button, and then click the "step 1" button, and a display like Figure 6 will appear.

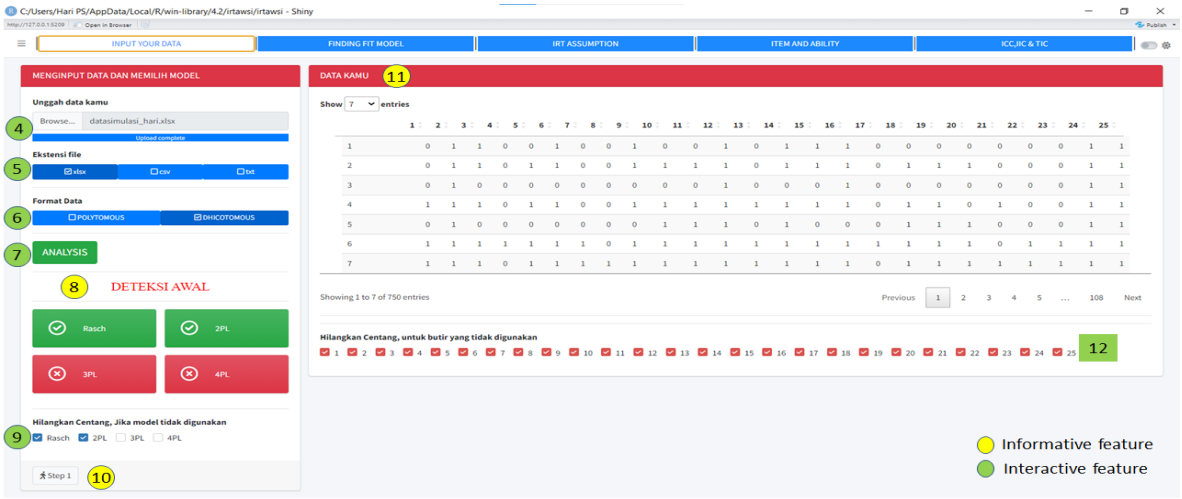


Sources: Personal data (2024).

Figure 6. Instructions for using the *irtawsi* package in layer 2.

Figure 6 shows that in layer two or step 1, the user is provided with a user manual to operate the package.

The illustration of using the *irtawsi* package in this article uses dichotomous data that can be downloaded from https://l1nk.dev/MOlgq.The data consists of 25 multiple-choice test items with 750 respondents and a file in .xlsx format. Following the usage instructions in Figure 6, the display in Figure 7 will appear.

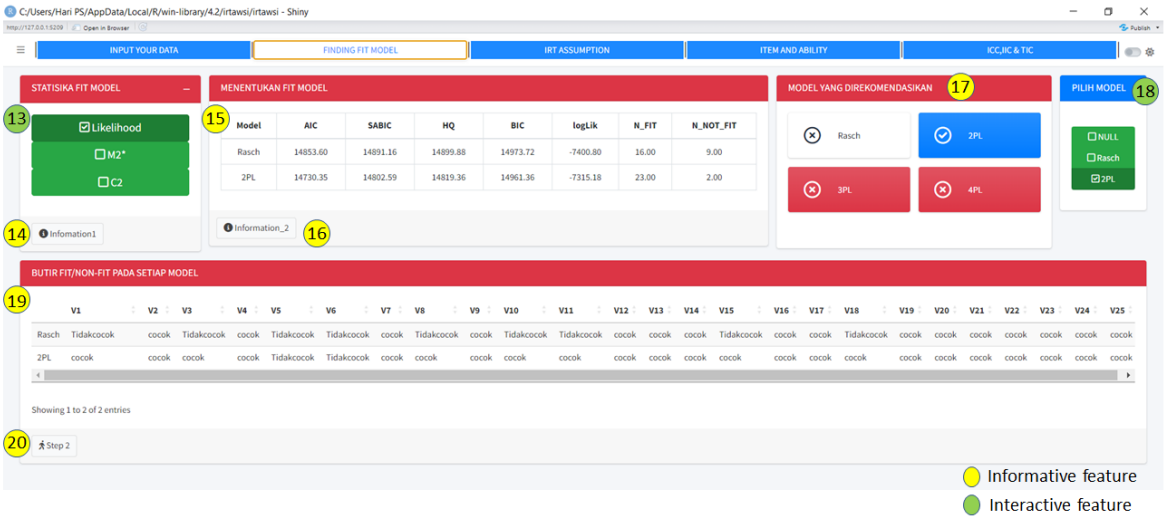


Sources: Personal data (2024).

Figure 7. The illustration of layer 2.

Figure 7 shows Layer 2 or Step 1. There are 12 features in this layer: (4) features for uploading data, (5) features for selecting the file extension used (in the illustration, the file has an Excel extension), (6) data format, which is adjusted to the type of data, either dichotomous or polytomous, (7) the Analysis button, which executes the initial detection, (8) initial detection, which provides information on the models that can or cannot be used, (9) a feature that continues the initial detection, (10) the Step 1 button, illustrated in Figure 6, (11) a feature that displays the analyzed data, and (12) a feature that eliminates one or more items that are not used in the analysis.

The analysis results in Figure 7 show that in feature (8), the 3PL and 4PL models are highlighted in red, indicating that these models cannot be used. Therefore, in feature (9), both models should be unchecked. Next, click the "Finding Fit Model" button to enter Layer 3 (Step 2), and the display will appear as shown in Figure 8.



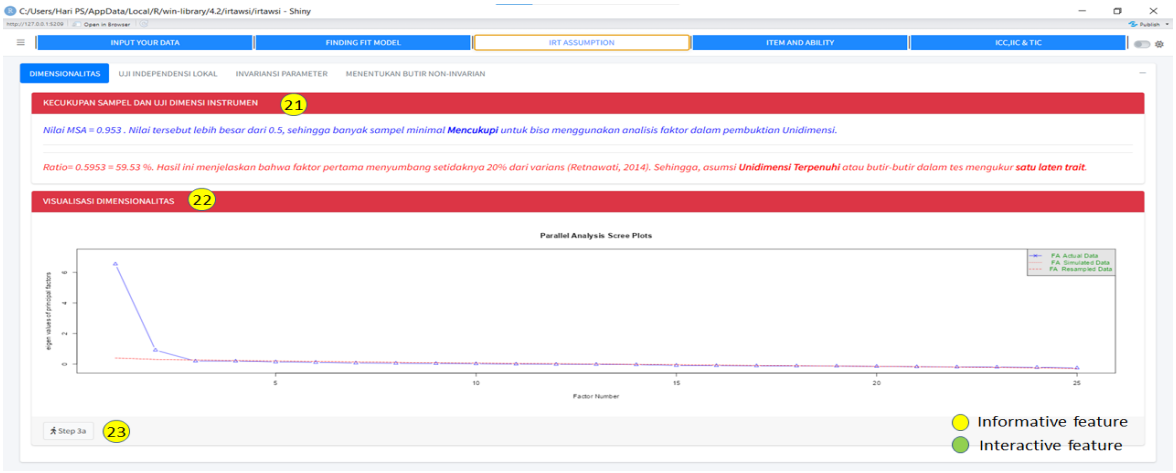
Sources: Personal data (2024).

Figure 8. The illustration of layer 3 in determining the Fit Model

Figure 8 shows Step 2. There are eight features in this layer: (13) the statistical method used to determine the best-fit model, (14) the Information1 button, which provides information on the method in feature 13, (15) a table comparing models, which can be used to manually determine the best-fit model, (16) the Information 2 button, which explains all information about the table in feature 15, (17) the recommended model, which can automatically recommend the best model, (18) selecting the model, the

main feature that the user must execute; if it is not executed, the analysis results in Layers 4 to 6 will not appear, (19) comparison of item fit in each model, and (20) the Step 2 button, which functions similarly to the Step 1 button (see Figure 6).

In the illustration in Figure 8, the statistical model used in feature (13) is likelihood, and the results show that the best-fit model is the 2PL model. Thus, the user can choose the 2PL model by selecting it in feature 18 and clicking the "IRT Assumption" button, resulting in the display as shown in Figure 9.

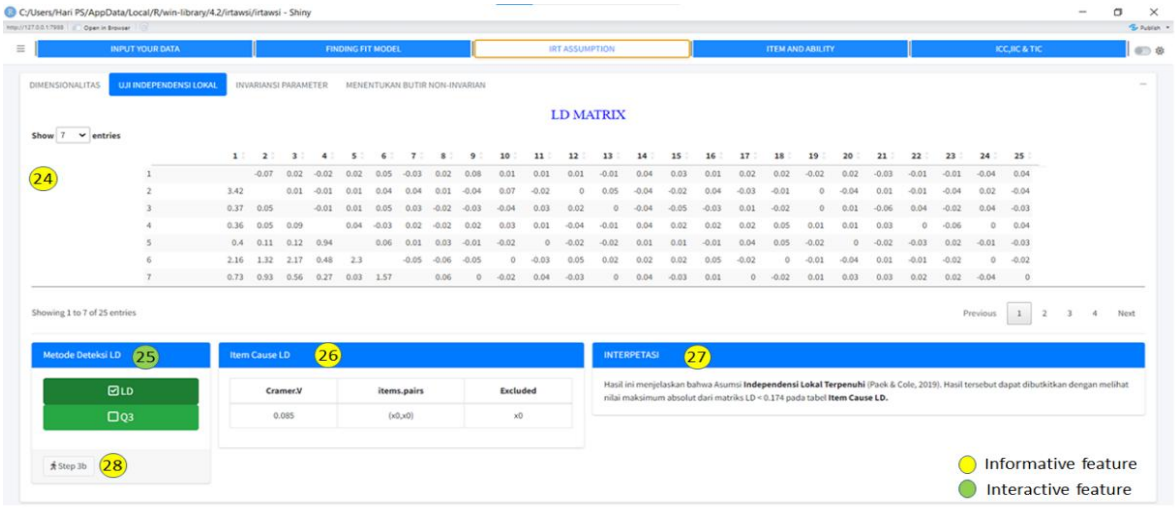


Sources: Personal data (2024).

Figure 9. The illustration of layer 4 in proving the unidimensional assumption.

Figure 9 illustrates the features of layer four or step 3a. There are three features in this section: (21) interpretation of unidimensionality assumption test results, (22) scree plot, which can be used by users who want to interpret the analysis results visually, and (23) Step 3a button.

The illustration in Figure 9 shows that the assumption test is met, so the test instrument used actually measures the same latent trait. Next, click the local independencies button, and a display like that in Figure 10 will appear.



Sources: Personal data (2024).

Figure 10. The illustration of layer 4 in proving the local independence assumption.

Figure 10 illustrates layer 4 for step 3b. There are 5 features in this layer: (24) LD matrix, which depends on feature 25. The upper triangular matrix on the LD matrix can be used manually to prove local independence. (25) Statistical methods for proving local independence, which feature 24 depends on. (26) Matrix cause LD, which is used to see which item pairs cause the assumption to be unmet. The

third column shows the item that should be deleted when there is local dependence. (27) Interpretation of the proof of local independence assumption. If this assumption is not met, this feature will automatically provide suggestions to the user. (28) Step 3b button provides information on layer 4 step 3b.

The illustration in Figure 10 shows that the LD method is used. The interpretation result explains that this assumption is met, so it can be continued to test the parameter invariance assumption. Next, click the invariance parameter button, and a display like that in Figure 11 will appear.

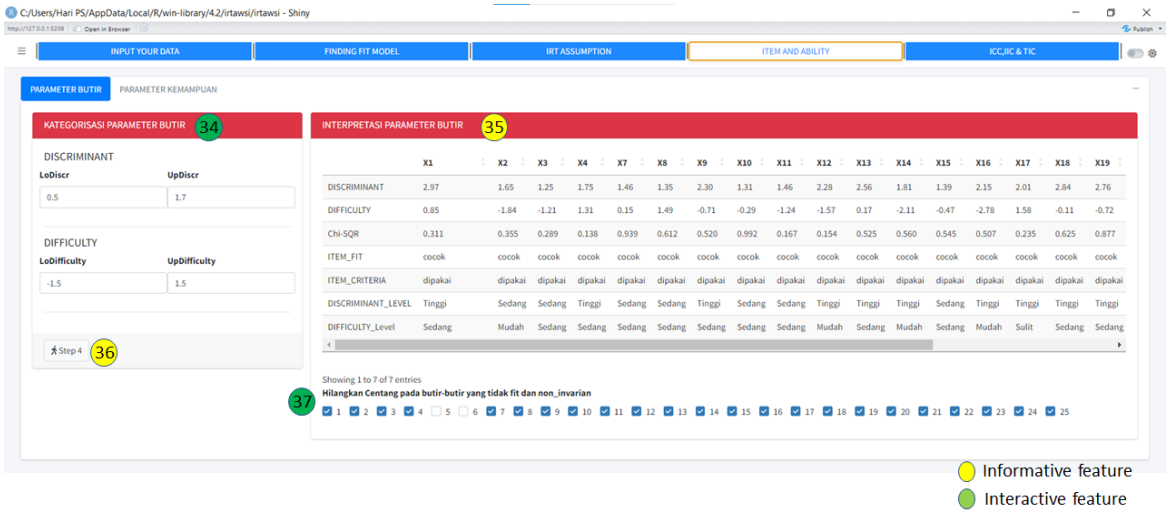
Sources: Personal data (2024).

Figure 11. The illustration of layer 4 in proving parameter invariant

Figure 11 illustrates layer 4 step 3c. There are five features in this layer: (29) Visual proof of parameter invariance. (30) Method for estimating ability parameters. The final image will change if this feature is changed to another method. (31) Feature for statistically proving parameter invariance. This feature is equipped with an interpretation of the analysis results for this assumption. (32) This feature helps users make decisions. If assumptions are not met in feature 31, then this feature will automatically provide suggestions regarding what the user should do.

The illustration in Figure 11 shows that the estimation method used for the parameter is EAP. The statistical analysis results from feature 31 explain that all parameter invariance assumptions are met. Based on these three assumptions tests, item parameter estimation can be performed, and item fit can be determined. Click the Item and Ability button, and a display like that in Figure 12 will appear.

http://journal.uinjkt.ac.id/index.php/jp3i

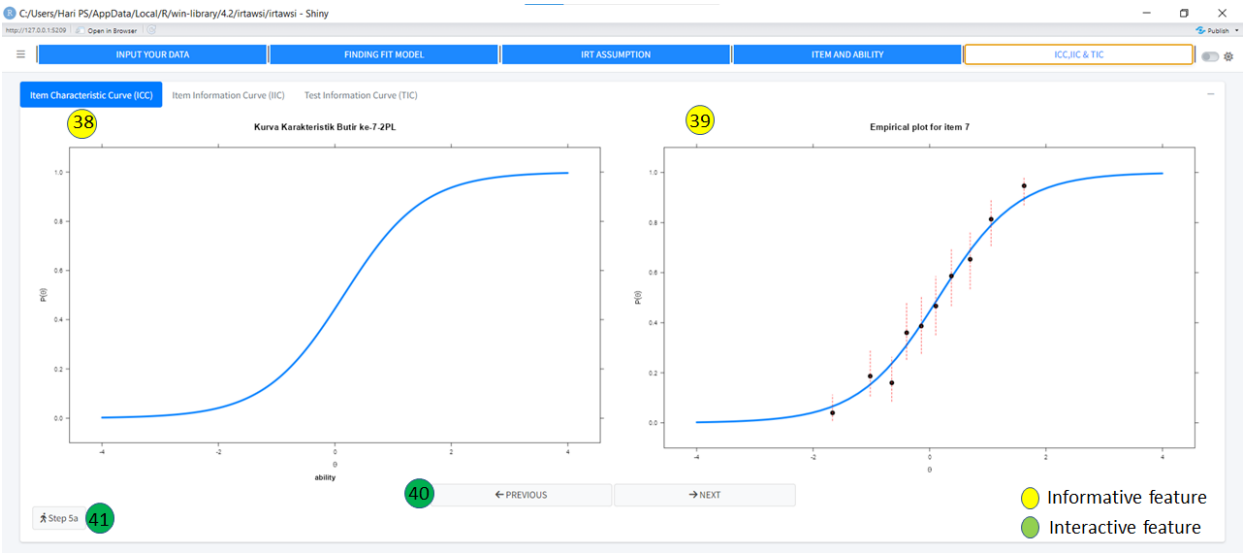


Sources: Personal data (2024).

Figure 12. The illustration of layer 5 in Fitting item and estimating item parameter

Figure 12 illustrates layer 5, step 4, which contains four features. (34) The item parameter categorization feature can be filled or left blank, depending on the reference used by the user. (35) The feature shows the estimated item parameters and the interpretation of fit and misfit. (36) The fourth feature contains information related to layer 5. (37) This feature unchecks the misfit or unused items.

The illustration in Figure 12 shows that items 5 and 6 are misfits, so they must be removed using feature 37. The IRT analysis process can be continued by pressing the ICC, IIC, and TIC buttons and clicking on the ICC button, displaying a view similar to that shown in Figure 13.

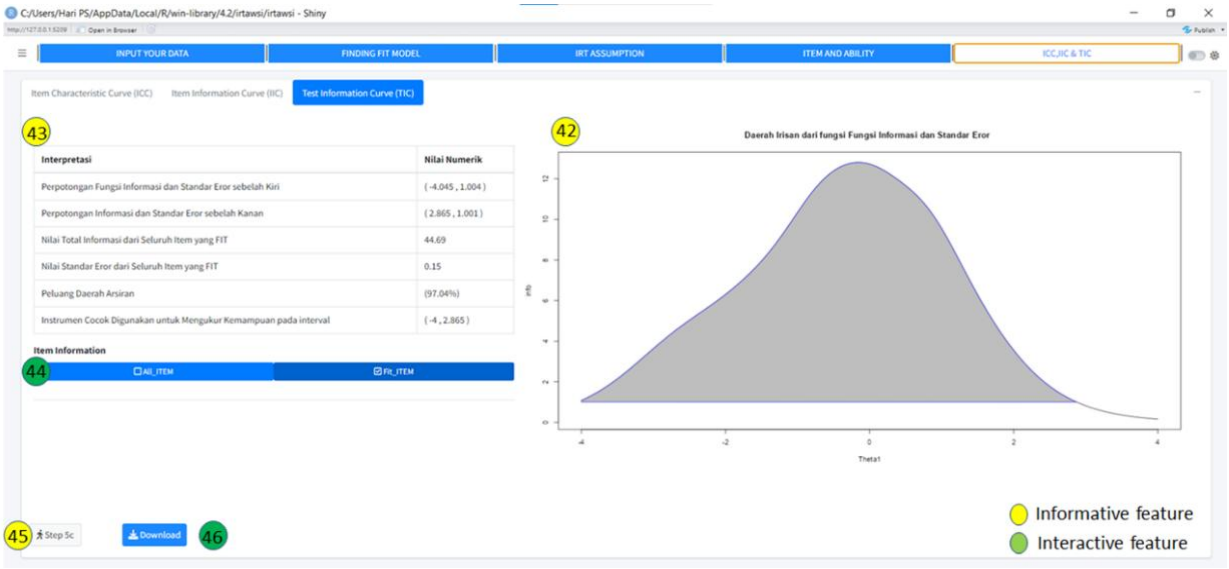


Sources: Personal data (2024).

Figure 13. Illustration of the display on layer 6 of the item characteristic curve (ICC)

Figure 13 shows layer 6 step 5. There are 4 features on this layer, (38) the item characteristic curve (ICC). (39) The item characteristic curve is accompanied by empirical values. (40) A feature to view the ICC of each item. (41) Step 5a button, containing information related to the ICC.

The illustration in Figure 13 explains that item 7 meets the monotonicity assumption. This result explains that the probability of an individual answering correctly increases as their ability improves. The IRT analysis process can be continued by pressing the TIC buttons, displaying a view similar to that shown in Figure 14**.**



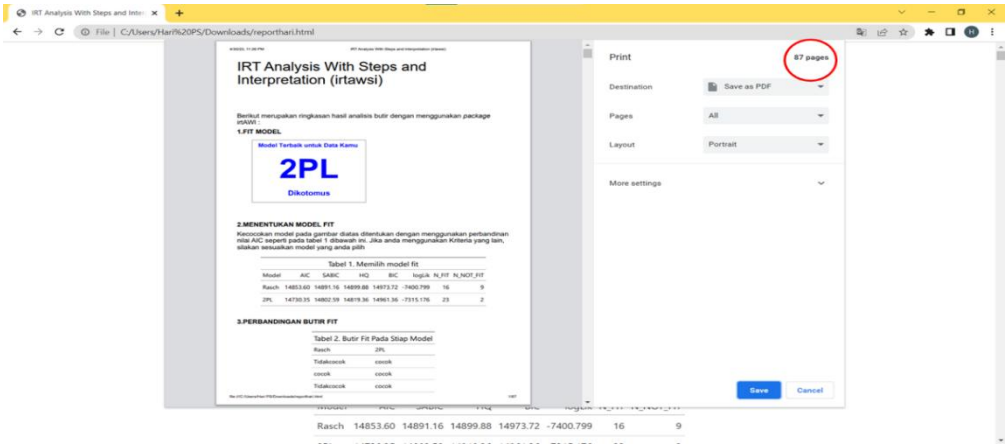
Sources: Personal data (2024).

Figure 14. The illustration of layer 6 Test Information Function (TIF)

Figure 14 shows layer 6 step 5, which contains five features: (42) the test information function curve, where shading indicates the area bounded by the TIF with SEM; (43) the interpretation of the figure in feature 38; (44) the feature to view the test information function for all items or only for fit items; (45) the step 5c button containing information related to layer 6, and (46) the download button that functions to download the analysis report document from the first step to the last step.

Based on feature 43 in Figure 14, the item information function value is 46.59, and the SEM value is 0.15. At the end of the table, it is shown that the developed test items are suitable for measuring student abilities within the range of -4 to 2.865. This interpretation indicates that the calibration process with *irtawsi* is complete. Click the download button to download the IRT analysis report document.

The analysis report is in html file format. Open the file to view the overall IRT analysis results. In the example provided in this article, if the html file is printed into a pdf file, it will result in an 87-page report, as shown in Figure 15.



Sources: Personal data (2024).

Figure 15. Conversion of HTML documents to PDF from the resulting IRT analysis

Figure 15 indicates that the user no longer needs to create reports manually. This demonstrates one of the advantages of the *irtawsi* package. For a more detailed illustration, please refer to https://youtu.be/DhZoXzbxiPs.

Discussion

The main capability of the *irtawsi* package is item analysis or calibration of test and non-test instruments using unidimensional IRT. This capability is realized through calibration steps (see Table 1). All calculations at each step are calculated with the *mirt* package, except for the unidimensional assumption. The unidimensional proof was determined with the psych package. These abilities are (1) Early detection. Initial detection is used to detect errors at the beginning of the calibration process. This error can occur because the data being analyzed does not have a maximum likelihood solution (Chalmers, 2012; Paek & Cole, 2019) or one or more items have the same response for each respondent. In the illustration, the 3PL and 4PL models are red. This shows that the analyzed data cannot be used for estimation in the 3PL and 4PL models. (2) Determine model fit. *irtawsi* can automatically recommend the most suitable model. However, if the user wants to determine manually, the user can see the results in the table provided (see Figure 8). (3) Test the IRT assumptions. If each assumption is met, irtawsi displays the analysis results and their interpretation (see Figure 9-11). Conversely, if the assumptions are not met. Then, Irtawsi will provide suggestions on how to overcome it. These suggestions can be read in the material section of this article. Subsequently, evaluate the monotonicity assumption. The case of items with dichotomous responses can be proven with ICC (see Figure 13). Meanwhile, in the case of polytomous items, *irtawsi* has not yet been used to detect the assumption of monotonicity. (4) Parameter estimation and item fit. The item parameter estimation results are displayed based on the selected model. If the user fills in the item parameter value limits, the user will automatically get the parameter categorization results. However, in this section, the interpretation of discriminant parameters in the interval 0.5 to 2 has not been executed automatically. This interval shows the ability of the items to differentiate students' abilities (Bichi & Talib, 2018; Hays et al., 2000). Apart from that, interpretation for other parameters has not been done automatically. Next, based on the chi-square value for each item, *irtawsi* automatically displays interpretations of fit and unfit items. (5) TIF and SEM. i*rtawsi* displays the TIF and SEM intersection areas. This area provides information that the calibrated test suits abilities at certain intervals (Retnawati, 2014; Susanto & Retnawati, 2023) (see Figure 14). These five capabilities are the main functionality of *irtawsi*, which packages with a user interface for IRT analysis do not have.

The functionalities of *irtawsi* have been tested through repeated verification processes and compared with the results from the basic statistical packages used (see Table 3). The overall results of the IRT analysis are consistent with manual calculations using the *mirt* package. Similarly, the unidimensionality test based on the FA parallel concept produces the same output as the *psych* package. Furthermore, the correlation analysis for verifying the assumption of parameter invariance also shows consistency with the *stats* package.

Apart from this main functionality, *irtawsi* also offers other functionalities, including (1) *irtawsi* can be used in two languages, namely English and Indonesian. (2) *irtawsi* can provide important information regarding the statistical methods used. This information is provided at almost every *irtawsi* layer. (3) *irtawsi* can write reports on analysis results and their interpretation from the first to the last step in html form.

The *irtawsi* functionality above cannot be activated without a user interface. This component is the main key to running every functionality of the package. *irtawsi* has a user interface that is not inferior to other packages. The *irtawsi* user interface has been adjusted based on the IRT analysis steps. In addition, the *irtawsi* user interface is equipped with an analysis guide, so users do not have to figure out how to use the package. The *irtawsi* user interface is equipped with interactive and informative parts, making it an application that is easy to remember, does not cause anxiety, and can be a medium for learning IRT for users who are learning IRT.

The functionality of *irtawsi* can be seen by comparing it with other packages developed based on the *R program*. The results of comparing *irtawsi* with the packages *irtGUI* (Yildiz, 2021), *IRTshiny* (Hamilton & Mizumoto, 2017), *catIRTtools* (Aybek, 2021) are shown in Table 4.

Table 4 explains the similarities and differences between *irtawsi* and other packages. These similarities and differences can be seen from several components. (1) basic package. *irtawsi*, *irtGUI*, and *catIRTtools* use the *mirt* package (Chalmers, 2012) for IRT analysis. *mirt* is one of the most complete *R program* packages for IRT analysis (Choi & Asilkalkan, 2019). On the other hand, *IRTshiny* uses ltm (Rizopoulos, 2006). (2) Language. *irtawsi* able to facilitate users using two languages. (3) IRT model. *irtawsi* can analyze PCM models. Meanwhile, other packages cannot. However, *irtawsi* does not have a nominal model, and RSM is like catIRTtools. (4), Calibration steps. In this component, *irtawsi* has many advantages. For example, detecting errors, being equipped with invariance assumptions, interpreting results, and providing suggestions is important. Next, in determining model fit, *irtawsi* offers two methods: the likelihood ratio test and the global fit methods. (5), person ability. In this component, *irtawsi* has the same capabilities as other packages. Respondent score estimates can only be calculated based on all items. In other words, both fit and unfit items are still used to estimate scores. This resulted in *irtawsi* being unable to estimate respondent scores outside the calibration sample. An example of estimated scores can be seen in research conducted by Susanto and Retnawati (2023). (6) IRT analysis guidelines. *irtawsi* facilitates analysis guidelines at each step of the analysis. This facility makes it easier for users without looking for software guidelines. (7) Information about IRT analysis. *irtawsi* and *irtGUI* both provide information related to the theory used. This information can be used as an initial reference for users to deepen their study of IRT.

Table 4. Differences and similarities between the *irtawsi* package and other packages.

Components	*Packages*			
	irtGUI	***IRTshiny***	***catIRTtools***	***irtawsi***
1. Basic package	*mirt*	*ltm*	*mirt*	*mirt*
2. Language	English	English	English	English and Indonesian
3. IRT model				
a. Rasch (1PL)	√	√	√	√
b. 2PL	√	√	√	√
c. 3PL	√	√	√	√
d. 4PL	√	√	√	√
e. GRM	√	√	√	√
f. PCM	–	–	–	√
g. GPCM	√	√	√	√
h. RSM	–	–	√	–
i. Nominal	–	–	√	–
4. Calibration Procedure				
a. Early detection	–	–	–	√
b. Determine the Fit model	√	√	√	√ **
c. Unidimensional assumption	√	√	√	√ **
d. local independence Assumption	√	–	–	√ **
e. Invariance assumption	–	–	–	√ **
f. Estimation of item parameters	√	√	√	√ *
g. fit item	√	√	√	√*
h. TIF	√	√	√	√ *
i. IIF	√	–	√	√
5. Person ability	√	√	√	√
6. IRT analysis guidelines	–	–	–	√
7. Information about IRT analysis	√	–	–	√

- The – symbol indicate that the component not present in the package
- The √ symbol indicate that the components are in the package
- The √ * *symbol* indicate that the components are in the package with the results interpretation
- The √ ** symbol indicate that the components are in the package with results interpretation and suggestions

Sources: Personal data (2024).

http://journal.uinjkt.ac.id/index.php/jp3i

Apart from the advantages above, *irtawsi* also has weaknesses. First, when the unidimensional assumption is not met, *Irtawsi* will provide suggestions for using multidimensional IRT. However, *irtawsi* does not provide these facilities. Second, *irtawsi* has not included an interpretation of the results in the ICC and IIF sections. Third, In this version (0.3.4) *irtawsi* cannot be used to estimate scores using only fit items. Apart from that, it is also not possible to estimate scores for respondents outside the sample.This weakness can be addressed with the 0.4.1 version. Fourth, *irtawsi* is not standalone software. *irtawsi* can be used when the user has installed the *R program.* Fifth, it cannot be used to detect monotonicity for polytomous cases.

The results of the development of *irtawsi* can be used as an alternative tool for IRT analysis. Interactive and informative features can make it easier for practitioners and researchers to calibrate instruments. Although some features still need to be completed.

Conclusion

Based on illustrations and discussion. *irtawsi* has a user interface equipped with the main functions of instrument calibration. *irtawsi* can also detect errors and tests of invariance assumptions, interpret analysis results, provide IRT analysis guidance, provide suggestions, and make reports on analysis results. It can also be used in two languages.

Further development will focus on several incomplete parts of the irtawsi. This package does not include interpretation regarding negative discriminant parameters, interpretation of ICC and IIC, monotonicity test for polytomous case, multidimensional IRT, and standalone version.

In summary, the results of this development contribute to the instrument calibration process. This package makes it easy for practitioners and researchers to calibrate instruments that are being developed. For beginner users, this package could be a learning medium for understanding the concept of unidimensional IRT. For IRT experts, this package is an alternative tool used for analysis.

Acknowledgment

The development of this package is part of a research project that has received funding support from the Ministry of Education, Culture, Research, and Technology (Kemendikbudristek) for the fiscal year 2024 under research agreements numbered 072/E5/PG.02.00.PL/2024 and T/105.1.46/UN34.9/PT.01.03/2024. We extend our sincere gratitude to Kemendikbudristek, Universitas Negeri Yogyakarta and STKIP PGRI Pacitan

Conflict of Interest

The authors declare no conflict of interest.

Authors Contribution

HPS: Conceptualization, Software, Visualization, Methodology, Writing – original draft, Writing – review & editing. *AMA*: Formal analysis, Investigation, review package. *H:* Conceptualization, Software, review package, Methodology. *HR*: Conceptualization, review package, Methodology, IRT conceptor. *RMA*: Conceptualization, Translate. *HD*: review package, IRT conceptor.

References

Ark, L. A. van der. (2007). Mokken Scale Analysis in R. *Journal of Statistical Software*, *20*(11). https://doi.org/10.18637/jss.v020.i11

Ark, L. A. van der. (2012). New Developments in Mokken Scale Analysis in R. *Journal of Statistical Software*, *48*(5). https://doi.org/10.18637/jss.v048.i05

Aybek, E. C. (2021). catIRT tools: A "Shiny" application for Item Response Theory calibration and computerized adaptive testing simulation. *Journal of Applied Testing Technology*, *22*(1).

Balafas, Spyros, E., Krijnen, Wim, P., Post, Wendy, J., & Wit, Ernst, C. (2020). mudfold: An R Package for Non-parametric IRT Modelling of Unfolding Processes. *The R Journal*, *12*(1), 49. https://doi.org/10.32614/RJ-2020-002

Battauz, M. (2020). Regularized Estimation of the Four-Parameter Logistic Model. *Psych*, *2*(4), 269–278. https://doi.org/10.3390/psych2040020

Bichi, A. A., & Talib, R. (2018). Item Response Theory: An Introduction to Latent Trait Models to Test and Item Development. *International Journal of Evaluation and Research in Education (IJERE)*, *7*(2), 142. https://doi.org/10.11591/ijere.v7i2.12900

Cavaliere, R. (2015). How to choose the right statistical software? - A method increasing the post-purchase satisfaction. *Journal of Thoracic Disease*, *7*(12). https://doi.org/10.3978/j.issn.2072-1439.2015.11.57

Chalmers, R. P. (2012). mirt : A Multidimensional Item Response Theory Package for the R Environment. *Journal of Statistical Software*, *48*(6). https://doi.org/10.18637/jss.v048.i06

Chen, W.-H., & Thissen, D. (1997). Local Dependence Indexes for Item Pairs Using Item Response Theory. *Journal of Educational and Behavioral Statistics*, *22*(3), 265–289. https://doi.org/10.3102/10769986022003265

Choi, Y.-J., & Asilkalkan, A. (2019). R Packages for Item Response Theory Analysis: Descriptions and Features. *Measurement: Interdisciplinary Research and Perspectives*, *17*(3), 168–175. https://doi.org/10.1080/15366367.2019.1586404

De Champlain, A. F. (2010). A primer on classical test theory and item response theory for assessments in medical education. *Medical Education*, *44*(1), 109–117. https://doi.org/10.1111/j.1365-2923.2009.03425.x

DeMars, C. (2010). Item Response Theory. In *Item Response Theory*. https://doi.org/10.1093/acprof:oso/9780195377033.001.0001

Dennis, A., Wixom, H. B., & Tegarden, D. (2015). Systems Analysis Design with UML Version 2.5: An Object-Oriented Approach. In *John Wiley & Sons*.

Edwards, M. C., Houts, C. R., & Cai, L. (2018). A Diagnostic Procedure to Detect Departures from Local Independence in Item Response Theory Models. *Psychological Methods*, *23*(1), 138–149. https://doi.org/10.1037/met0000121

Foster, G. C., Min, H., & Zickar, M. J. (2017). Review of Item Response Theory Practices in Organizational Research. *Organizational Research Methods*, *20*(3), 465–486. https://doi.org/10.1177/1094428116689708

Granjon, D. (2022). *bs4Dash: A "Bootstrap 4" Version of "shinydashboard"* (R package version 2.2.1). https://cran.r-project.org/package=bs4Dash

Guenole, N., & Brown, A. (2014). The consequences of ignoring measurement invariance for path coefficients in structural equation models. *Frontiers in Psychology*, *5*. https://doi.org/10.3389/fpsyg.2014.00980

Hackenberger, B. K. (2020). R software: unfriendly but probably the best. *Croatian Medical Journal*, *61*(1), 66–68. https://doi.org/10.3325/cmj.2020.61.66

Hambleton, R., Swaminathan, H., & Rogers, H. J. (1991). *Fundamental of Item Response Theory*. SAGE.

Hamilton, W. K., & Mizumoto, A. (2017). *IRTShiny: Item Response Theory via Shiny*. https://cran.r-project.org/package=IRTShiny

Han, K. (Chris) T., & Paek, I. (2014). A Review of Commercial Software Packages for Multidimensional

IRT Modeling. *Applied Psychological Measurement*, *38*(6), 486–498. https://doi.org/10.1177/0146621614536770

Hays, R. D., Morales, L. S., & Reise, S. P. (2000). Item Response Theory and health outcomes measurement in the 21st century. *Medical Care*, *38*(9 SUPPL. 2). https://doi.org/10.1097/00005650-200009002-00007

Hori, K., Fukuhara, H., & Yamada, T. (2020). Item Response Theory and Its Applications in Educational Measurement Part I: Item Response Theory and Its Implementation in R. *WIREs Computational Statistics*, *14*(2). https://doi.org/10.1002/wics.1531

Hu, Y., & Plonsky, L. (2021). Statistical assumptions in L2 research: A systematic review. *Second Language Research*, *37*(1), 171–184. https://doi.org/10.1177/0267658319877433

Iannone, R., Cheng, J., Schloerke, B., Hughes, E., & Seo, J. (2022). *Package gt* (p. 306).

Jumailiyah, M. (2017). Item response theory: A basic concept. *Educational Research and Reviews*, *12*(5), 258–266. https://doi.org/10.5897/ERR2017.3147

Lameijer, C. M., van Bruggen, S. G. J., Haan, E. J. A., Van Deurzen, D. F. P., Van der Elst, K., Stouten, V., Kaat, A. J., Roorda, L. D., & Terwee, C. B. (2020). Graded Response Model Fit, Measurement Invariance and (Comparative) Precision of the Dutch-Flemish PROMIS® Upper Extremity V2.0 Item Bank in Patients with Upper Extremity Disorders. *BMC Musculoskeletal Disorders*, *21*(1), 170. https://doi.org/10.1186/s12891-020-3178-8

Magis, D., & Barrada, J. R. (2017). Computerized Adaptive Testing with R : Recent Updates of the Package catR. *Journal of Statistical Software*, *76*(Code Snippet 1). https://doi.org/10.18637/jss.v076.c01

Nguyen, T. H., Han, H.-R., Kim, M. T., & Chan, K. S. (2014). An Introduction to Item Response Theory for Patient-Reported Outcome Measurement. *The Patient - Patient-Centered Outcomes Research*, *7*(1), 23–35. https://doi.org/10.1007/s40271-013-0041-0

Paek, I., & Cole, K. (2019). *Using R for Item Response Theory Model Applications*. Routledge. https://doi.org/10.4324/9781351008167

Paura, L., & Arhipova, I. (2012). Advantages and Disadvantages of Professional and Free Software for Teaching Statistics. *Information Technology and Management Science*, *15*(1). https://doi.org/10.2478/v10313-012-0001-z

Perrier, V., Meyer, F., & Granjon, D. (2023). *shinyWidgets: Custom Inputs Widgets for Shiny* (R package version 0.7.6). https://cran.r-project.org/package=shinyWidgets

Petersen, M. A. (2005). Introduction to Non-parametric Item Response Theory. *Quality of Life Research*, *14*(4), 1201–1202. https://doi.org/10.1007/s11136-005-1259-7

Posit team. (2023). *RStudio: Integrated Development Environment for R*. http://www.posit.co/

R Core Team. (2022). *R: A Language and enviroment for statiscital computing.* R Foundation for statistical Computing. https://www.R-project.org/.

R Core Team. (2023). *R: A Language and Environment for Statistical Computing*. https://www.r-project.org/

Reeve, B. B., Hays, R. D., Bjorner, J. B., Cook, K. F., Crane, P. K., Teresi, J. A., Thissen, D., Revicki, D. A., Weiss, D. J., Hambleton, R. K., Liu, H., Gershon, R., Reise, S. P., Lai, J., & Cella, D. (2007). Psychometric Evaluation and Calibration of Health-Related Quality of Life Item Banks. *Medical Care*, *45*(5), S22–S31. https://doi.org/10.1097/01.mlr.0000250483.85507.04

Reise, S. P., Du, H., Wong, E. F., Hubbard, A. S., & Haviland, M. G. (2021). Matching IRT Models to Patient-Reported Outcomes Constructs: The Graded Response and Log-Logistic Models for Scaling Depression. *Psychometrika*, *86*(3), 800–824. https://doi.org/10.1007/s11336-021-09802-0

Reise, S. P., Moore, T. M., & Haviland, M. G. (2013). Applying unidimensional item response theory

models to psychological data. In K. F. Geisinger, B. A. Bracken, J. F. Carlson, J.-I. C. Hansen, N. R. Kuncel, S. P. Reise, & M. C. Rodriguez (Eds.), *APA handbook of testing and assessment in psychology, Vol. 1: Test theory and testing and assessment in industrial and organizational psychology.* (pp. 101–119). American Psychological Association. https://doi.org/10.1037/14047-006

Retnawati, H. (2014). *Teori Respons Butir dan Penerapannya untuk Peneliti, Praktisi Pengukuran dan Pengujian, Mahasiswa Pascasarjana.* Nuha Medika.

Retnawati, H. (2015). Karakteristik Butir Tes dan Analisisnya. *Uny*, *53*(5).

Retnawati, H. (2016). *Validitas , Reliabilitas dan Karakteristik Butir* (1st ed.). Parama Publising.

Rizopoulos, D. (2006). ltm : An R Package for Latent Variable Modeling and Item Response Theory Analyses. *Journal of Statistical Software*, *17*(5). https://doi.org/10.18637/jss.v017.i05

Sali, A., & Attali, D. (2020). *shinycssloaders: Add Loading Animations to a "shiny" Output While It's Recalculating* (R package version 1.0.0). https://cran.r-project.org/package=shinycssloaders

Shatz, I. (2023). Assumption-checking rather than (just) testing: The importance of visualization and effect size in statistical diagnostics. *Behavior Research Methods*. https://doi.org/10.3758/s13428-023-02072-x

Sijtsma, K., & van der Ark, L. A. (2022). Advances in non-parametric item response theory for scale construction in quality-of-life research. *Quality of Life Research*, *31*(1), 1–9. https://doi.org/10.1007/s11136-021-03022-w

Soetaert, K. (2020). *diagram: Functions for Visualising Simple Graphs (Networks), Plotting Flow Diagrams* (R package version 1.6.5). https://cran.r-project.org/package=diagram

Sudaryono. (2013). *Toeri Responsi Butir* (pertama). Graha Ilmu.

Susanto, H. P., & Retnawati, H. (2023). Kalibrasi Instrumen Literasi Matematika Siswa Menggunakan IRT dan Aplikasinya untuk Estimasi Skor. *Edumatica: Jurnal Pendidikan Matematika*, *13*(1), 23–36. https://doi.org/10.22437/edumatica.v13i01.23135

Susanto, H. P., Retnawati, H., Abadi, A. M., Haryanto, H., Ali, R. M., & Djidu, H. (2024). *irtawsi : Items Response Theory Analysis with Steps and Interpretation* (0.4.1). CRAN. https://doi.org/https://doi.org/10.32614/CRAN.package.irtawsi

Tilley, S., & Rosenblatt, H. (2016). Systems Analysis and Design, Eleventh Edition. In *A Guide to Medical Computing*.

Toland, M. D. (2014). Practical Guide to Conducting an Item Response Theory Analysis. *The Journal of Early Adolescence*, *34*(1), 120–151. https://doi.org/10.1177/0272431613511332

Wickham, H., & Bryan, J. (2023). *readxl: Read Excel Files* (R package version 1.4.2). https://cran.r-project.org/package=readxl

Wickham, H., Bryan, J., & Barrett, M. (2022). *usethis: Automate Package and Project Setup*. https://cran.r-project.org/package=usethis

William Revelle. (2023). *psych: Procedures for Psychological, Psychometric, and Personality Research* (R package version 2.3.3). https://cran.r-project.org/package=psych

Xie, Y., Cheng, J., & Tan, X. (2023). *DT: A Wrapper of the JavaScript Library "DataTables"* (R package version 0.27). https://cran.r-project.org/package=DT

Xie, Y., Dervieux, C., & Riederer, E. (2020). *R Markdown Cookbook*. Chapman and Hall/CRC. https://bookdown.org/yihui/rmarkdown-cookbook

Xu, J., Zhang, Q., & Yang, Y. (2020). Impact of Violations of Measurement Invariance in Cross-Lagged Panel Mediation Models. *Behavior Research Methods*, *52*(6), 2623–2645. https://doi.org/10.3758/s13428-020-01426-z

Yen, W. M. (1984). Effects of Local Item Dependence on the Fit and Equating Performance of the Three-Parameter Logistic Model. *Applied Psychological Measurement*, *8*(2). https://doi.org/10.1177/014662168400800201

Yildiz, H. (2021). *irtGUI: Item Response Theory Analysis with a Graphic User Interface*. CRAN. https://doi.org/10.32614/CRAN.package.irtGUI