
MENGUKUR KREATIFITAS DAN KUALITAS PEMOGRAMAN PADA SISWA SMK KOTA PEKANBARU JURUSAN TEKNIK KOMPUTER JARINGAN DENGAN SIMULASI ROBOT

Anip Febtriko ¹⁾, Ira Puspitasari ²⁾

¹⁾²⁾ Teknik Informatika, Fakultas Teknik, Universitas Abdurrah Pekanbaru
Jl. Riau Ujung, Pekanbaru, 28282
Alamat Email: aniep.febtric@gmail.com, irapuspitasari@gmail.com

ABSTRAK

Mempelajari pemrograman sebaiknya dimulai dari cara-cara yang sederhana dan menyenangkan, terutama untuk seorang siswa. Apabila diawal dikenalkan dengan cara-cara yang sulit dipahami dan terkesan membosankan maka akan mengurangi minat belajar siswa. Kemudian jika dipaksakan untuk belajar di khawatirkan nantinya akan menghasilkan sistem yang tidak maksimal. Meskipun siswa mempunyai algoritma untuk menyelesaikan suatu permasalahan yang berbeda-beda tapi akan lebih baik kita gunakan algoritma yang lebih ringkas dan sederhana. Target silmulasi ini adalah SMK yang ada di kota Pekanbaru. Untuk mengukur ketercapaian kegiatan penelitian menggunakan scale likert dimana Skala Likert adalah skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti. Skala ini merupakan suatu skala psikometrik yang biasa diaplikasikan dalam angket dan paling sering digunakan untuk riset yang berupa survei, termasuk dalam penelitian survei deskriptif.

Kata Kunci: Skala Likert, Robomind, Algoritma

ABSTRACT

Learning programming should start from simple and fun ways, especially for a student. If the beginning is introduced in ways that are difficult to understand and seem boring then it will reduce the student's interest in learning. Then if forced to learn in the future will result in a system that is not maximal. Although students have algorithms to solve a different problem but it would be better if we use a more concise and simple algorithm. The target of this silmulasi is the existing SMK in Pekanbaru city. To measure research activity activity using Likert scale where Likert Scale is the scale used to measure perception, attitude or opinion of a person or group about an event or social phenomenon, based on operational definition which has been determined by researcher. This scale is a psychometric scale commonly applied in questionnaires and most often used for research in the form of surveys, including in descriptive survey research.

Keywords: likert Scale, Robomind, Algorithms

I. ENDAHULUAN

1. Latar Belakang Masalah

Saat ini masih banyak sekali siswa mengalami kendala di dalam memahami pemograman. Kendala-kendala tersebut diantaranya mengenai alur logika penyelesaian sistem tersebut. Sebelum membuat suatu pemograman siswa terlebih dahulu harus mengetahui langkah-langkah atau alur proses logika pemograman. Kesalahan di dalam merumuskan alur logika dapat berakibat sistem yang dihasilkan nanti kurang bagus dan bisa

mendatangkan masalah. Masalah tersebut diantaranya alur penyelesaian sistemnya terlalu panjang dan sebagainya. Maka untuk itulah perumuskan algoritma memegang peranan yang sangat penting di dalam pembuatan sebuah sistem pemograman.

Mempelajari pemograman sebaiknya dimulai dari cara-cara yang sederhana dan menyenangkan, terutama untuk seorang siswa. Apabila diawal dikenalkan dengan cara-cara yang sulit dipahami dan terkesan membosankan maka akan mengurangi minat belajar siswa. Kemudian jika dipaksakan untuk

belajar di khawatirkan nantinya akan menghasilkan sistem yang tidak maksimal.

Meskipun siswa mempunyai algoritma untuk menyelesaikan suatu permasalahan yang berbeda-beda tapi akan lebih baik kita gunakan algoritma yang lebih ringkas dan sederhana.

Maka untuk menumbuhkan kualitas dan kreatifitas siswa dalam pemrograman bisa menggunakan alternatif simulasi robot supaya mendapatkan kesan belajar lebih menyenangkan dan tidak membosankan.

II. TINJAUAN PUSTAKA

1. Pengertian Program

Pada umumnya program adalah sederetan instruksi atau statement yang tentunya dalam bahasa yang dimengerti oleh komputer. Instruksi tersebut berfungsi untuk mengatur pekerjaan apa saja yang akan dilakukan oleh komputer agar mendapatkan dan menghasilkan suatu hasil atau keluaran yang diharapkan (Budiyanto & Algoritma, 2003).

Menurut (P & Somantri, n.d.) kata program dapat diartikan:

1. Untuk mendeskripsikan instruksi-instruksi tersendiri, yang biasanya disebut *source code*, yang dibuat *programmer*.
2. Untuk mendeskripsikan suatu keseluruhan bagian dari *software* yang *executable*.

Dapat juga dikatakan bahwa sebuah program merupakan himpunan atau kumpulan instruksi tertulis yang dibuat oleh *programmer* atau suatu bagian *executable* dari sebuah *software*.

Pengertian (Peningkatan et al., 2016) Pemrograman menurut merupakan

“kegiatan menulis kode program yang akan dieksekusi oleh komputer”.

Menurut (Chalid, Informatika, Industri, & Gunadarma, 2009), bahasa pemrograman adalah “perangkat lunak atau *software* yang dapat digunakan dalam proses pembuatan program yang melalui beberapa tahapan-tahapan penyelesaian masalah”.

Proses pemrograman komputer bukan saja sekedar menulis suatu urutan instruksi yang harus dikerjakan oleh komputer akan tetapi bertujuan untuk memecahkan suatu masalah serta membuat mudah pekerjaan pengguna komputer (*user*). Didalam membuat sebuah program komputer, tentu tidak terlepas dari sifat individu pemrogram (*Programmer*). Karakteristik seorang pemrogram yang mutlak dimiliki yaitu:

- a. Memiliki pola pikir yang logis
- b. Memiliki ketekunan dan ketelitian yang tinggi
- c. Memiliki penguasaan bahasa pemrograman yang baik
- d. Memiliki pengetahuan teknik pemrograman yang baik.

Untuk membuat suatu Program yang kompleks tahap-tahap yang harus dilakukan *programmer* adalah :

1. Definisi Masalah
Programmer harus memahami permasalahan yang timbul kemudian mengidentifikasi permasalahan yang ada sehingga dapat menentukan batasan masalah.
2. Analisa Kebutuhan
Programmer harus menentukan kebutuhan data untuk masukan dan keluaran yang diminta, bahasa pemrograman yang digunakan serta

- tipe komputer apa sebagai pendukungnya.
3. Desain Algoritma
Algoritma yang didesain harus memiliki kebenaran secara logika sebelum siap diimplementasikan ke dalam bentuk program.
 4. Bahasa Pemrograman
Bahasa Pemrograman adalah media untuk membuat Program.
 5. *Testing* dan *Debugging*
Testing untuk menguji program sampai terbebas dari kesalahan. *Debugging* untuk mengoreksi kesalahan yang terdeteksi. Ada tiga macam kesalahan yang biasanya terjadi :
 - a. *Syntax Error*
Bentuk kesalahan program yang terjadi karena kesalahan dalam hal penulisan instruksi di dalam program.
 - b. *Run Time Error*
Bentuk kesalahan program yang terjadi karena adanya proses aritmatik yang tidak dapat diproses.
 - c. *Logical Error*
Bentuk kesalahan yang terjadi karena kesalahan logika program yang dibuat oleh *programmer*.
 6. Dokumentasi digunakan untuk *file* cadangan (*Backup*)
 7. Pemeliharaan
Yaitu dalam upaya menghindari kerusakan atau hilangnya suatu program baik dari *Factor Software* (Virus Program) ataupun dari *Factor Brainware* (*Human Error*).
Adapun jenis bahasa pemrograman berdasarkan perkembangannya :
 1. Bahasa Mesin (*Machine Language*)
Bahasa pemrograman yang hanya dapat dimengerti oleh mesin (komputer) yang di dalamnya terdapat CPU yang hanya mengenal dua keadaan yang berlawanan yaitu :
 - a. Bila terjadi kontak atau ada arus bernilai 1
 - b. Bila tidak ada kontak atau arus bernilai 0
 2. Bahasa Tingkat Rendah (*Low Level Language*)
Karena banyak keterbatasan yang dimiliki bahasa mesin maka dibuatlah symbol yang mudah diingat yang disebut dengan *Mnemonic* (pembantu untuk mengingat). Contoh : Bahasa *Assembler* yang dapat menerjemahkan *Mnemonic*.
 3. Bahasa Tingkat Menengah (*Middle Level Language*)
Bahasa pemrograman yang menggunakan aturan-aturan gramatikal dalam penulisan pernyataannya, mudah untuk dipahami, dan memiliki instruksi-instruksi tertentu yang dapat langsung diakses oleh komputer.
Contoh : Bahasa C
 4. Bahasa Tingkat Tinggi (*High Level Language*)
Bahasa pemrograman yang dalam penulisan pernyataannya mudah dipahami secara langsung.
 - a. Bahasa Berorientasi pada Prosedur (*Procedure Oriented Language*)
Contoh : Algoritma, Fortran, Pascal, Basic, Cobol, PL/1
 - b. Bahasa Berorientasi pada Masalah (*Problem Oriented Language*)

Contoh : RPG (*Report Program Generator*)

5. Bahasa Berorientasi Obyek (*Object Oriented Language*)

Bahasa pemrograman ini mengandung fungsi-fungsi untuk menyelesaikan suatu permasalahan dan *programmer* tidak harus menulis secara detail semua pernyataannya, tetapi cukup memasukan kriteria-kriteria yang dikehendaki saja.

Contoh : Visual dBase, Visual FoxPro, Delphi, Visual C, Visual Basic.

Terdapat tiga faktor penting dalam bahasa pemrograman yaitu :

1. Sintaks
Sintaks merupakan aturan penulisan atau tata bahasa pemrograman.
2. Semantik
Semantik adalah arti atau maksud yang terkandung di dalam setiap *statement* dalam sebuah program.
3. Kebenaran Logika
Kebenaran Logika berhubungan dengan benar atau tidaknya urutan *statement* dalam sebuah program.

Di dalam suatu pemrograman harus memiliki kerangka dasar pemrograman, yaitu suatu proses pengolahan data yang terdiri dari tiga tahapan dasar yang disebut dengan Siklus Pengolahan Data (*Data Processing Cycle*) yang terdiri dari *Input*, *Processes* dan *Output*.

Tiga tahapan tersebut dapat dikembangkan lebih lanjut dengan

menambah beberapa tahapan lagi sebagai berikut :

1. *Originating*
Tahap ini berhubungan dengan proses pengumpulan data yang biasanya merupakan proses pencatatan (*recording*) data ke dokumen dasar.
2. *Input*
Tahap ini merupakan proses pemasukan data (*entry data*) ke dalam proses komputer melalui peralatan *Input (Input Device)*.
3. *Process*
Tahap ini merupakan proses pengolahan data yang sudah dimasukan, berupa proses menghitung, membandingkan, mengklasifikasikan, mengurutkan, mengendalikan dan mencari di *Storage*.
4. *Output*
Tahap ini merupakan proses yang menghasilkan keluaran dari proses pengolahan data melalui peralatan *output (Output Device)* berupa informasi.
5. *Distribution*
Tahap ini merupakan proses penyebaran informasi kepada pihak-pihak yang berhak dan membutuhkan informasi.
6. *Storage*
Tahap ini merupakan perekaman hasil pengolahan data ke *Secondary Storage*, yang dapat dipergunakan sebagai bahan *input* untuk proses selanjutnya.

Dalam penerapannya, program adalah salah satu bentuk dari berbagai macam jenis aplikasi yang dipergunakan

dalam bidang bisnis ataupun dalam ilmu pengetahuan (*science*), untuk menghasilkan suatu bentuk laporan atau tujuan yang diinginkan.

2. Scale Likert

Skala Likert adalah skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti. Skala ini merupakan suatu skala psikometrik yang biasa diaplikasikan dalam angket dan paling sering digunakan untuk riset yang berupa survei, termasuk dalam penelitian survei deskriptif (Qomari, 2008).

Penggagas dan pencipta skala likert adalah Rensis Likert asal Amerika Serikat yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial. Dengan Skala Likert, variabel yang akan diukur dijabarkan menjadi indikator variabel.

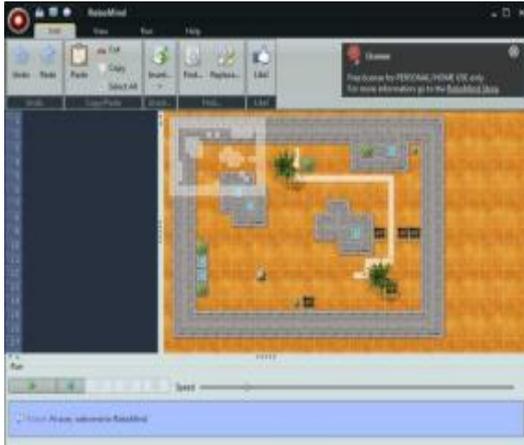
Dalam pengukuran bidang pendidikan, skala Likert juga sering digunakan, selain juga skala Guttman, semantik Diferensial, Rating scale, dan skala Thurstone . Dalam penggunaan skala Likert, terdapat dua bentuk pertanyaan, yaitu bentuk pertanyaan positif untuk mengukur skala positif, dan bentuk pertanyaan negatif untuk mengukur skala negatif. Pertanyaan positif diberi skor 5, 4, 3, 2, dan 1; sedangkan bentuk pertanyaan negati diberi skor 1, 2, 3, 4, dan 5 atau -2, -1, 0, 1, 2. Bentuk jawaban skala Likert antara lain: sangat setuju, setuju, ragu-ragu, tidak setuju, dan tidak setuju.

Selain itu, jawaban setiap item instrumen yang menggunakan Skala Likert bisa juga mempunyai gradasi dari sangat positif sampai sangat negatif, yang dapat berupa kata-kata antara lain: Sangat Penting (SP), Penting (P), Ragu-ragu (R), Tidak Penting (TP), Sangat Tidak Penting (STP).

3. RoboMind

Robomind merupakan salah satu perangkat lunak yang dikembangkan khusus untuk keperluan pendidikan terutama di bidang teknologi simulasi robot. Robomind dikembangkan oleh Arvid Halma, seorang siswa dari University of Amsterdam pada waktu itu. Sejak 2011 RoboMind diterbitkan oleh Reseach Kitchen. Robomind merupakan sebuah simulasi robot game yang didalamnya kita bisa belajar struktur pemrograman seperti halnya pemrograman pada umumnya, diantaranya seperti bentuk perulangan, pemilihan dan prosedur (Nugroho, Tinggi, & Bontang, 2015).

Dengan simulator ini nantinya akan kita gunakan untuk belajar algoritma pemrograman. Pada saat kita ketikkan script pada console maka kita akan bisa langsung tahu seperti apa hasil dari script yang kita ketikkan sebelumnya. Robomind merupakan satu diantara banyaknya game simulator yang ada. Tetapi diantara simulator-simulator yang lain, yang paling sederhana dan mudah untuk digunakan berlatih algoritma yaitu robomind (Yuana & Maryono, 2016). Karena perintah-perintah yang dipakai pada aplikasi tersebut masih sangat sederhana dan sangat mudah untuk dipakai bahkan untuk anak-anak. Tampilan aplikasi robomind dapat dilihat pada gambar 1.

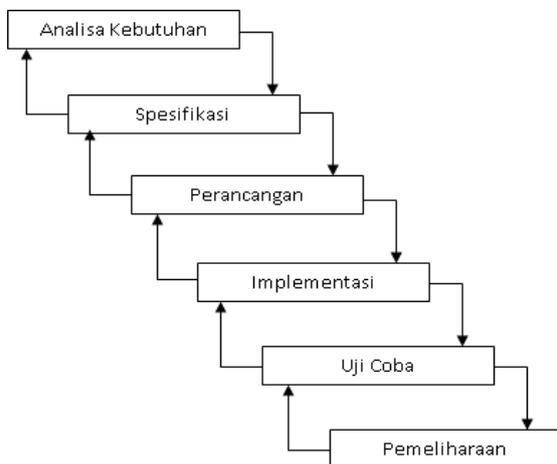


Gambar 1. Tampilan simulator robomind.

III. METODE

1. Kerangka Kerja Penelitian

Agar mempermudah dalam pengerjaan penelitian, maka peneliti membuat kerangka kerja penelitian seperti pada gambar 3.1 berikut ini.



Gambar 3.1 Kerangka Kerja Penelitian

Pada gambar 3.1 merupakan kerangka kerja yang digunakan dalam langkah-langkah penelitian.

2. Analisis Kebutuhan

Tahap analisis kebutuhan merupakan tahap awal pada suatu penelitian tahap ini dilakukan dengan melihat latar belakang dari penelitian, kemudian melihat batasan masalah dan selanjutnya melakukan perancangan.

3. Spesifikasi

Spesifikasi adalah pengumpulan data merupakan langkah kedua dalam melakukan penelitian. Data dikumpulkan dari berbagai sumber yang ada.

Pengumpulan data pada penelitian ini dilakukan dengan 2 metode, yaitu metode wawancara, studi pustaka (*literature*).

1. Wawancara

Wawancara adalah proses komunikasi atau interaksi untuk mengumpulkan informasi dengan cara tanya jawab antara peneliti dengan informan atau subjek.

3. Studi kepustakaan (*library research*)

Melakukan Penelitian di perpustakaan dengan cara mencari buku dan jurnal kemudian mempelajari literatur yang ada hubungannya dengan topik yang diteliti.

4. Perancangan sistem

Merupakan tahap penelitian proses data, aliran proses dan hubungan antar data yang paling optimal dan memenuhi kebutuhan pihak yang sesuai dengan hasil analisa kebutuhan.

1. Perancangan Proses

Rancangan proses menggunakan *Diagram Alir Data* (DAD) pada sistem deteksi jarak pada pengawasan tempat penitipan anak.

2. Perancangan Flowchart

Bagan alir program (*flowchart*) adalah suatu bagan yang menggambarkan arus logika dari data yang akan diproses

dalam suatu program dari awal sampai akhir.

5. Implementasi dan Pengujian Sistem

Pada tahap ini peneliti mengimplementasikan sistem berdasarkan rancangan yang telah dibuat sebelumnya sesuai dengan kebutuhan pihak terkait. Dan tahap pengujian terhadap sistem yang telah dibuat sekaligus mengevaluasi kekurangan serta kelebihan sistem tersebut.

6. Pemeliharaan

Tahap ini dilakukan guna mengevaluasi sistem yang telah dibuat guna memperoleh hasil yang optimal, dan perawatan terhadap sistem bertujuan agar sistem yang sudah terpasang dapat berjalan semakin efektif dan efisien, perawatan sistem baru harus dilakukan secara berturut turut dan terencana. Apabila program terdapat *error*, maka akan diadakan perbaikan pada program dengan cara memperbaiki dan menghilangkan eror yang terjadi sampai program dapat digunakan dengan baik dan sempurna.

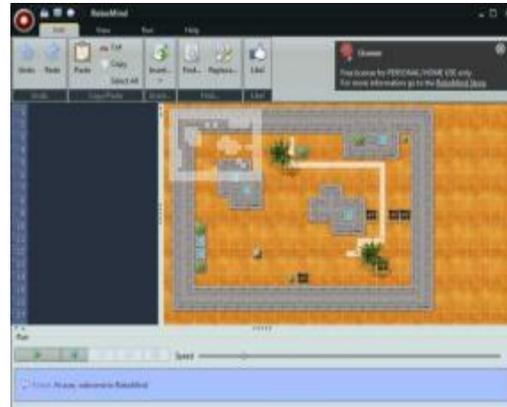
7. Alat dan Bahan

Pada penelitian ini menggunakan fasilitas komputer dalam melakukan analisa dan perancangan. Adapun spesifikasi komputer yang digunakan adalah sebagai berikut :

1. Hardware
 - a. Prosesor
 - b. Memory
 - c. Monitor
 - d. VGA Ouboard
2. Software
 - a. Sistem Operasi
 - b. Bahasa pemrograman
 - c. RoboMind

IV. HASIL DAN PEMBAHASAN

1. Simulasi Program Pengenalan RoboMind



Gambar 4.1.1. Tampilan simulator robomind.

2. Algoritma RoboMind

a. if

```
right
forward(3)
left
forward(1)
paintWhite
forward(3)
if(frontIsClear){right}
forward(1)
backward(2)
stopPainting
backward(2)
```

Gambar a. Algoritma If

b. Repeat

```
right
forward(8)
repeat
{
  if(frontIsWhite){
    forward(1)
  }
  else if(rightIsWhite){
    right
  }
  else if(leftIsWhite){
```

Gambar b. Algoritma Repeat

3. Pengukuran Scale Likert

Scale Likert

- SS = Sangat Setuju
- S = Setuju
- R = Ragu
- TS = Tidak Setuju
- STS = Sangat Tidak Setuju

Jumlah Responden

Jumlah responden sebanyak 106 dengan 3 sekolah yang ada di kota pekanbaru

4. Hasil Pembahasan

1. Pertanyaan

Pertanyaan dengan 1 variable (x) yakni :

Apakah Bahasa Pemrograman dengan silmulasi Robomind dapat dipahami ?

2. Hasil jawaban

Tabel Hasil Jawaban

Kategori	Skor
SS	43
S	32
R	12
TS	13
STS	6
Total	106

3. Tabel Bobot Nilai

Tabel Bobot Nilai

A	5
B	4
C	3
D	2
E	1

4. Presentasi Nilai

Tabel Presentase Nilai

Jawaban	Keterangan
0% - 19.99%	Sangat (Tidak Setuju, Buruk atau Kurang Sekali)
20% -	Tidak Setuju atau

39.99%	Kurang Baik
40% - 59.99%	Cukup atau Netral
60% - 79.99%	Setuju, Baik atau Suka
80% - 100%	Sangat (Setuju, Baik, Suka)

Dari data yang didapat diatas kemudian diolah dengan cara mengkalikan setiap point jawaban dengan bobot yang sudah ditentukan dengan tabel bobot nilai.

Maka Hasil Perhitungan jawaban responden sebagai berikut :

- 1) Responden yang menjawab sangat setuju

$$(5) = 43 \times 5 = 20$$

- 2) Responden yang menjawab setuju

$$(4) = 32 \times 4 = 28$$

- 3) Responden yang menjawab netral

$$(3) = 12 \times 3 = 51$$

- 4) Responden yang menjawab tidak setuju

$$(2) = 13 \times 2 = 16$$

- 5) Responden yang menjawab tidak sangat setuju

$$(1) = 6 \times 1 = 0$$

$$\text{Total Skor} = 215 + 128 + 36 + 26 + 6 = 411$$

Untuk mendapatkan hasil interpretasi, harus diketahui dulu skor tertinggi (X) dan angka terendah (Y) untuk item penilaian dengan rumus sebagai berikut :

Y = Skor tertinggi likert x jumlah responden (Angka Tertinggi 5) "Perhatikan Bobot Nilai" X = Skor terendah likert x jumlah responden (Angka Terendah 1) "Perhatikan Bobot Nilai"

Jumlah skor tertinggi untuk item **SANGAT SETUJU** ialah $5 \times 106 = 530$,

sedangkan item **SANGAT TIDAK SETUJU** ialah $1 \times 106 = 106$. Jadi, jika total skor responden di peroleh angka 411, maka penilaian interpretasi responden terhadap Silmulasi Pemrograman tersebut adalah hasil nilai yang dihasilkan dengan menggunakan rumus Index %.

Rumus Index % = Total Skor / Y x 100

Maka penyelesaian akhir dari contoh kasus

:

= Total Skor / Y x 100

= $411/530 \times 100$

= $77.54\% = 77\%$ Kategori **SETUJU**

Dari hasil di atas maka dapat disimpulkan bahwa responden **SETUJU Bahasa Pemrograman dengan silmulasi Robomind** dapat dipahami.

V. KESIMPULAN

1. Kesimpulan

1. Penerapan silmulasi bahasa pemrograman kepada siswa dapat dan mampu memahami siswa terhadap bahasa atau alur algoritma yang diberikan dalam bentuk silmulasi.
2. Pengukuran dengan menggunakan scale likert dapat menjadi alat ukur ketercapaian silmulasi yang diberikan kepada siswa dengan indikator angka yang tepat dan akurat.

DAFTAR PUSTAKA

- [1] Budiyanto, a., & algoritma, a. I. (2003). *Pengantar algoritma dan pemrograman*, 1–5.
- [2] Chalid, i. R., informatika, j. T., industri, f. T., & gunadarma, u. (2009). Aplikasi audio steganografi untuk melindungi data, 1–7.
- [3] Nugroho, a., tinggi, s., & bontang, t. (2015). Pengenalan algoritma pemrograman melalui simulasi robot, 2015(november), 1–7.
- [4] P, a. B., & somantri, m. (n.d.). Perancangan aplikasi e-education berbasis web.
- [5] Peningkatan, u., konsep, p., algoritma, d., robomind, m. A., model, m., based, p., ... nofitasari, a. (2016). Surakarta juni 2016.
- [6] Qomari, r. (2008). Pengembangan instrumen evaluasi domain afektif, 13(1), 1–15.
- [7] Yuana, r. A., & maryono, d. (2016). Robomind utilization to improve student motivation and concept in learning programming, 1, 962–966.