



APLIKASI PENGHITUNG PERSOALAN TEORI HIMPUNAN BERBASIS ANDROID DENGAN BAHASA PEMROGRAMAN KOTLIN

Hendro Purwoko¹, Bayu Jaya Tama²

¹²Program Studi Teknik Informatika, Universitas Indraprasta PGRI, Indonesia

Corresponding Author:

Bayu Jaya Tama,
Program Studi Teknik Informatika,
Universitas Indraprasta PGRI,
Jl. Nangka Raya No.58 C, Jakarta Selatan, Jakarta, Indonesia.
Email: bayujaya88@gmail.com
Contact Person: 0856-9282-4900

Informasi Artikel:

Submite 19 Desember, 2021
Direvisi 2 Januari, 2022
Diterima 19 Januari, 2022

How to Cite:

Hendro Purwoko, Bayu Jaya Tama. Aplikasi Penghitung Persoalan Teori Himpunan Berbasis Android Dengan Bahasa Pemrograman Kotlin. *Jurnal Theorems (The Original Research of Mathematics, X(X))*, 01-21.

ABSTRAK

Matematika dapat dikombinasikan dengan Teknologi pada bidang Sistem Informasi karena kesamaannya berpikir dengan logika. Kombinasi tersebut diterapkan dalam membahas Teori Himpunan dengan membuat aplikasi berbasis Android menggunakan bahasa pemrograman Kotlin. Tahap pembuatan aplikasi menggunakan SDLC (Software Development Life Cycle) bertipe Waterfall agar setiap tahapan dilakukan dengan baik dan melakukan pengujian dengan membuat beberapa simulasi agar setiap perhitungan menghasilkan nilai yang benar. Hasil keluaran dari penelitian menghasilkan aplikasi penghitung Teori Himpunan berbasis Android dengan perhitungan yang akurat.

Kata kunci: Teori Himpunan, Android, Kotlin, Waterfall

ABSTRACT

Mathematics can be combined with Technology in the field of Information Systems because of its similarities to thinking with logic. This combination is applied in discussing Set Theory by creating Android-based applications using the Kotlin programming language. The application development stage uses the Waterfall type SDLC (Software Development Life Cycle) so that each stage is carried out properly and performs testing by making several simulations so that each calculation produces the correct value. The output of the research results in an Android-based Set Theory calculator application with accurate calculations.

Keywords: Set Theory, Android, Kotlin, Waterfall

PENDAHULUAN

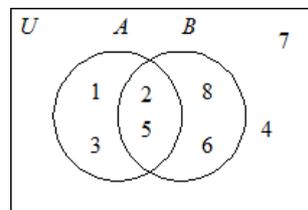
Matematika merupakan salah satu cabang ilmu pengetahuan dan memegang peranan penting dalam pendidikan, karena matematika merupakan metode berpikir ilmiah yang diperlukan untuk mengembangkan kemampuan nalar, logika, dan salah satu mata pelajaran yang memberikan kontribusi positif dalam tercapainya warga yang cerdas dan bermartabat melalui sikap kritis dan berpikir logis. Namun pada kenyataannya, matematika masih dianggap sebagai materi yang sulit dipahami bagi siswa (Riana & Ibrahim, 2019).

Konsep matematika dapat dikombinasikan bersamaan dengan teknologi pada bidang Sistem Informasi dikarenakan memiliki kesamaan dalam berpikir secara logis, misalnya dalam bahasan teori

Himpunan. Teori himpunan merupakan teori yang paling dasar bagi cabang ilmu matematika, disadari atau tidak, dalam kehidupan sehari-hari, sesungguhnya kita telah mengetahui dan banyak menerapkan konsep himpunan.

Himpunan merupakan kumpulan benda atau objek yang didefinisikan dengan jelas (Hanifah, 2020), menurut Darwanto (2020) Objek yang terdapat dalam himpunan disebut elemen, unsur, atau anggota. Contoh kumpulan yang bukan himpunan dalam pengertian matematika adalah: (1) kumpulan nilai; (2) kumpulan mobil bagus; dan (3) kumpulan musik merdu. Pada contoh tersebut tampak bahwa dalam suatu kumpulan ada objek. Objek tersebut dapat abstrak atau dapat juga konkret. Menyatakan suatu himpunan dapat dengan cara deskripsi yaitu cara menyatakan suatu himpunan dengan kata-kata dan cara tabulasi yaitu cara menyatakan himpunan dengan cara mendaftar anggota himpunan satu persatu (Vandini, 2015)

Metode penyajian himpunan menggunakan diagram Venn (Darwanto, 2020). Diagram venn menyajikan himpunan dalam bentuk grafis. Misalkan $U = \{1, 2, 3, 4, 5, 6, 7, 8\}$ atau dapat ditulis $U = \{1, 2, \dots, 7, 8\}$, $A = \{1, 2, 3, 5\}$, dan $B = \{2, 5, 6, 8\}$ dapat dibuat diagram Venn sebagai berikut:

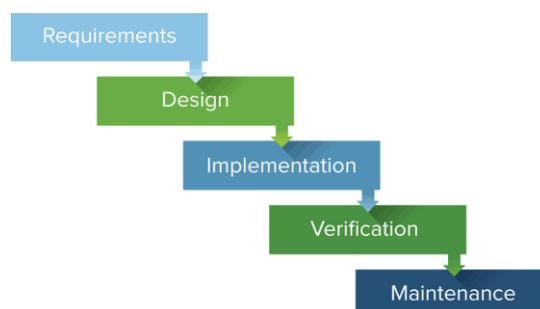


Gambar 1. Diagram Venn

Berdasarkan bahasan diatas maka dibuatlah aplikasi untuk melakukan perhitungan Teori Himpunan berbasis Android dengan menggunakan bahasa pemrograman Kotlin pada perangkat lunak Android Studio dan menerapkan SDLC (Software Development Life Cycle) bertipe Waterfall agar aplikasi yang dibentuk lebih teratur mengikuti tahapan yang berurut.

METODE PENELITIAN

Dalam melakukan pembuatan aplikasi Teori Himpunan menggunakan Metode atau dalam pengembangan perangkat lunak dikenal dengan SDLC (Software Development Life Cycle) bertipe Waterfall dengan urutan seperti tergambar dibawah ini.



Gambar 2. Alur SDLC *Waterfall*

Dengan detail tahapan sesuai pada tabel 1.

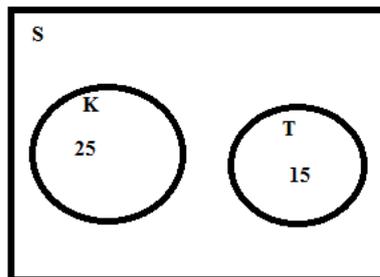
Tabel 1. Detail kegiatan pada SDLC *Waterfall*

Tahap	Kegiatan
<i>Requirements</i>	Melakukan diskusi khusus membahas tentang Teori Himpunan dengan hingga mendapati algoritma yang sesuai untuk diterapkan saat melakukan pengodean dengan menggunakan bahasa pemrograman Kotlin berbasis Android
<i>Design</i>	Merancang desain sistem menggunakan <i>use case</i> pada UML dan rancangan antar-muka dengan Figma
<i>Implementation</i>	Melakukan penerapan dengan cara menulis kode menggunakan bahasa pemrograman Kotlin dengan perangkat lunak Android Studio
<i>Verification</i>	Pengujian dengan membuat simulasi
<i>Maintenance</i>	Memperbaiki dan meningkatkan versi aplikasi yang telah dibuat

HASIL DAN PEMBAHASAN

Dalam membangun aplikasi Teori Himpunan perlu membuat simulasi perhitungan sebagai berikut:

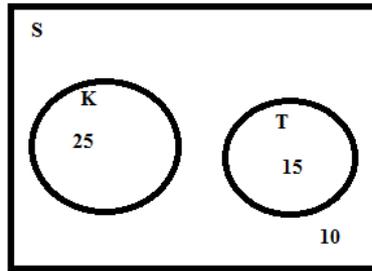
1. Dalam sebuah kelas, terdapat 25 mahasiswa suka minum kopi, 15 mahasiswa suka minum teh. Berapa banyak mahasiswa dalam kelas itu?



Gambar 3. Diagram Venn Simulasi Pertama

$$n(S) = 25 + 15 = 40$$

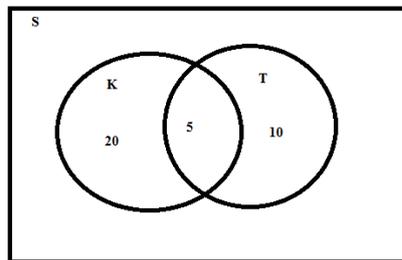
2. Dalam sebuah kelas, terdapat 25 mahasiswa suka minum kopi, 15 mahasiswa suka minum teh dan 10 mahasiswa tidak suka keduanya. Berapa banyak mahasiswa dalam kelas itu?



Gambar 4. Diagram Venn Simulasi Kedua

$$N(s) = 25 + 15 + 10 = 50$$

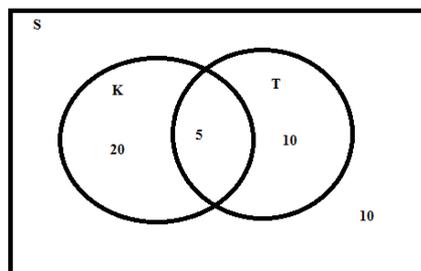
3. Dalam sebuah kelas, terdapat 25 mahasiswa suka minum kopi, 15 mahasiswa suka minum teh dan 5 mahasiswa suka keduanya. Berapa banyak mahasiswa dalam kelas itu?



Gambar 5. Diagram Venn Simulasi Ketiga

$$N(s) = 20 + 5 + 10 = 35$$

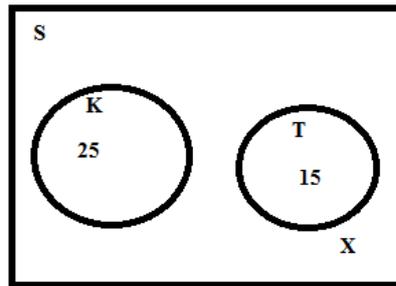
4. Dalam sebuah kelas, terdapat 25 mahasiswa suka minum kopi, 15 mahasiswa suka minum teh, 5 mahasiswa suka keduanya, dan 10 mahasiswa tidak suka keduanya. Berapa banyak mahasiswa dalam kelas itu?



Gambar 6. Diagram Venn Simulasi Keempat

$$N(S) = 20 + 5 + 10 + 10 = 45$$

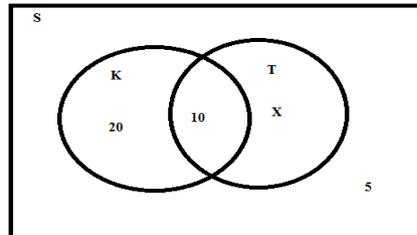
5. Dalam sebuah kelas yang berjumlah 40 mahasiswa, terdapat 25 mahasiswa suka minum kopi, 15 mahasiswa suka minum teh, Berapa banyak mahasiswa yang tidak suka minum keduanya?



Gambar 7. Diagram Venn Simulasi Kelima

$$\begin{aligned} N(s) &= 25 + 15 + X \\ 40 &= 40 - X \\ X &= 0 \end{aligned}$$

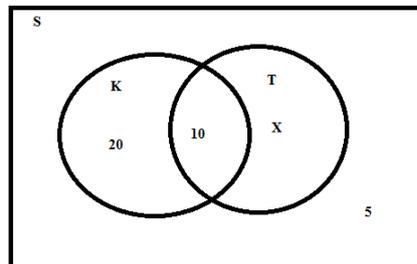
6. Dalam sebuah kelas yang berjumlah 40 mahasiswa, terdapat 30 mahasiswa suka minum kopi, 10 mahasiswa suka minum kopi dan teh, dan 5 mahasiswa tidak suka keduanya. Berapa banyak mahasiswa yang suka minum teh?



Gambar 8. Diagram Venn Simulasi Keenam

$$\begin{aligned} N(s) &= 20 + 10 + X + 5 \\ 40 &= 35 + X \\ X &= 40 - 35 = 5 \\ \text{Teh} &= 10 + 5 = 15 \end{aligned}$$

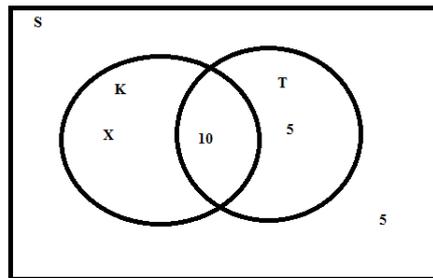
7. Dalam sebuah kelas yang berjumlah 40 mahasiswa, terdapat 30 mahasiswa suka minum kopi, 10 mahasiswa suka minum kopi dan teh, dan 5 mahasiswa tidak suka keduanya. Berapa banyak mahasiswa yang suka minum teh saja?



Gambar 9. Diagram Venn Simulasi Ketujuh

$$\begin{aligned} N(s) &= 20 + 10 + X + 5 \\ 40 &= 35 + X \\ X &= 40 - 35 = 5 \text{ (suka teh saja)} \end{aligned}$$

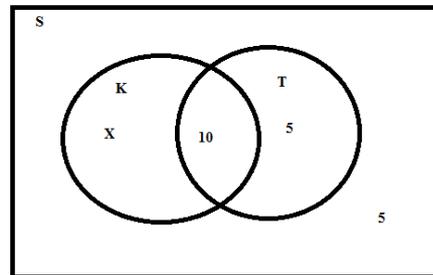
8. Dalam sebuah kelas yang berjumlah 40 mahasiswa, terdapat 15 mahasiswa suka minum teh, 10 mahasiswa suka minum kopi dan teh, dan 5 mahasiswa tidak suka keduanya. Berapa banyak mahasiswa yang suka minum kopi?



Gambar 10. Diagram Venn Simulasi Kedelapan

$$\begin{aligned} N(s) &= X + 10 + 5 + 5 \\ 40 &= X + 20 \\ X &= 40 - 20 = 20 \\ \text{Kopi} &= 20 + 10 = 30 \end{aligned}$$

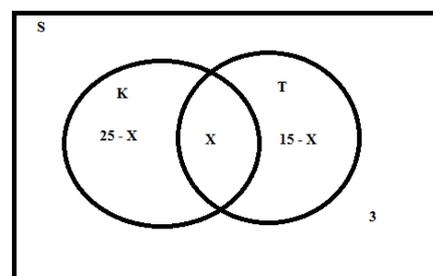
9. Dalam sebuah kelas yang berjumlah 40 mahasiswa, terdapat 15 mahasiswa suka minum teh, 10 mahasiswa suka minum kopi dan teh, dan 5 mahasiswa tidak suka keduanya. Berapa banyak mahasiswa yang suka minum kopi saja?



Gambar 11. Diagram Venn Simulasi Kesembilan

$$\begin{aligned} N(s) &= X + 10 + 5 + 5 \\ 40 &= X + 20 \\ X &= 40 - 20 = 20 \text{ (kopi saja)} \end{aligned}$$

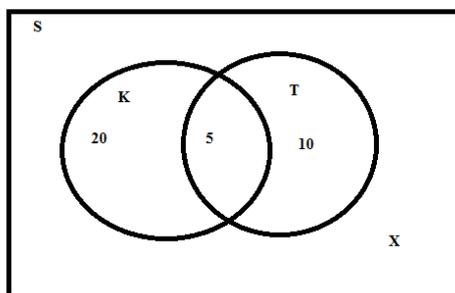
10. Dalam sebuah kelas yang berjumlah 38 mahasiswa, terdapat 25 mahasiswa suka minum kopi, 15 mahasiswa suka minum teh, dan 3 mahasiswa tidak suka keduanya. Berapa banyak mahasiswa yang suka minum keduanya?



Gambar 12. Diagram Venn Simulasi Kesepuluh

$$\begin{aligned} N(S) &= (25 - X) + X + (15 - X) + 3 \\ 38 &= 43 - X \\ X &= 43 - 38 = 5 \end{aligned}$$

11. Dalam sebuah kelas yang berjumlah 38 mahasiswa, terdapat 25 mahasiswa suka minum kopi, 15 mahasiswa suka minum teh, dan 5 mahasiswa suka keduanya. Berapa banyak mahasiswa yang tidak suka minum keduanya?



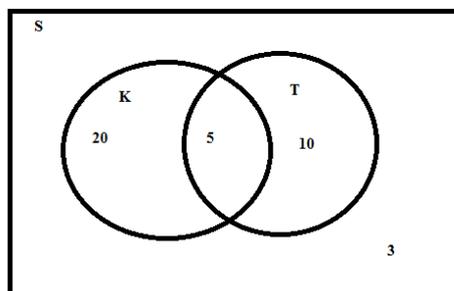
Gambar 13. Diagram Venn Simulasi Kesebelas

$$N(S) = 20 + 5 + 10 + X$$

$$38 = 35 + X$$

$$X = 38 - 35 = 3$$

12. Dalam sebuah kelas yang berjumlah 38 mahasiswa, terdapat 25 mahasiswa suka minum kopi, 15 mahasiswa suka minum teh, 5 mahasiswa suka keduanya, dan 3 mahasiswa tidak suka keduanya. Berapa banyak jumlah mahasiswa yang suka minum kopi dan minum teh?



Gambar 14. Diagram Venn Simulasi Kedua Belas

Jumlah yang minum kopi dan teh
 $= 20 + 5 + 10 = 35$

Atau

Jumlah yang minum kopi dan teh $= 38 - 3 = 35$

Simulasi tersebut diterapkan pada aplikasi dengan hasil uji coba pada Tabel 2.

Tabel 2. Simulasi

Simulasi	Semesta	A	B	C	\cap	Temukan	Hasil	Sesuai
1	0	25	15	0	0	S	40	Ya
2	0	25	15	10	0	S	50	Ya
3	0	25	15	0	5	S	35	Ya
4	0	25	15	10	5	S	45	Ya
5	40	25	15	0	0	K	0	Ya
6	40	30	0	5	10	B	15	Ya
7	40	30	0	5	10	B Saja	5	Ya
8	40	0	5	5	10	A	30	Ya
9	40	0	5	5	10	A Saja	20	Ya
10	38	25	15	0	3	\cap	5	Ya

11	38	20	10	0	5	C	3	Ya
12	38	25	10	3	5	Gabungan	35	Ya

Berdasarkan simulasi diatas kode pemrograman berbahasa Kotlin ditulis sebagai berikut

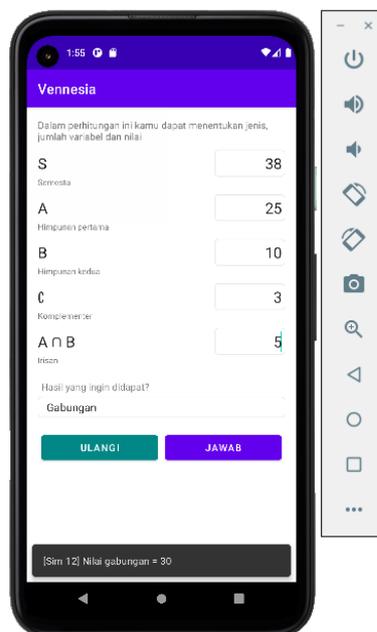
```
binding.btAnswer.setOnClickListener {
    var s = Integer.parseInt(binding.variableSemesta.text.toString());
    var a = Integer.parseInt(binding.variable1.text.toString());
    var b = Integer.parseInt(binding.variable2.text.toString());
    var c = Integer.parseInt(binding.variable3.text.toString());
    var i = Integer.parseInt(binding.variable4.text.toString());
    val g: Int

    if (NILAI == 0){
        if (a != 0 && b != 0 && c != 0 && i == 0 && s != 0) {
            c = (s - i)
            c = (a + b) - c
            Snackbar.make(binding.rootLayout, "Nilai S = "+ c.toString(),
                Snackbar.LENGTH_SHORT).show()
        } else if (a != 0 && b != 0 && c == 0 && i == 0 && s == 0) {
            s = a + b
            Snackbar.make(binding.rootLayout, "[Sim 1] Nilai S = "+
                s.toString(), Snackbar.LENGTH_SHORT).show()
        } else if (a != 0 && b != 0 && c != 0 && i == 0 && s == 0) {
            s = a + b + c
            Snackbar.make(binding.rootLayout, "[Sim 2] Nilai S = "+
                s.toString(), Snackbar.LENGTH_SHORT).show()
        } else if (a != 0 && b != 0 && c == 0 && i != 0 && s == 0) {
            s = (a - i) + (b - i) + i
            Snackbar.make(binding.rootLayout, "[Sim 3] Nilai S = "+
                s.toString(), Snackbar.LENGTH_SHORT).show()
        } else if (a != 0 && b != 0 && c != 0 && i != 0 && s == 0) {
            s = (a - i) + (b - i) + i + c
            Snackbar.make(binding.rootLayout, "[Sim 4] Nilai S = "+
                s.toString(), Snackbar.LENGTH_SHORT).show()
        } else {
            s = s - i
            Snackbar.make(binding.rootLayout, "[Sim 12] Nilai S = "+
                s.toString(), Snackbar.LENGTH_SHORT).show()
        }
    } else if (NILAI == 1){
        a = ((s - b) - c) + i
        Snackbar.make(binding.rootLayout, "[Sim 8] Nilai A = "+
            a.toString(), Snackbar.LENGTH_SHORT).show()
    } else if (NILAI == 2){
        a = (s - b) - c
        Snackbar.make(binding.rootLayout, "[Sim 9] Nilai A Only = "+
            a.toString(), Snackbar.LENGTH_SHORT).show()
    } else if (NILAI == 3){
        if (a != 0 && b == 0 && c != 0 && i != 0 && s != 0) {
            //simulasi 6
            b = ((s - c) - a) + i
            Snackbar.make(binding.rootLayout, "[Sim 6] Nilai B = "+
                b.toString(), Snackbar.LENGTH_SHORT).show()
        }
    } else if (NILAI == 4){
        if (a != 0 && b == 0 && c != 0 && i != 0 && s != 0) {
            b = ((s - c) - a) + i
            Snackbar.make(binding.rootLayout, "[Sim 7] Nilai B Only = "+
                b.toString(), Snackbar.LENGTH_SHORT).show()
        }
    } else if (NILAI == 5){ //c maksudnya komplementer
        if (a != 0 && b != 0 && c == 0 && i == 0 && s != 0) {
```

```

        c = s - ( ( a + b ) - i )
        Snackbar.make(binding.rootLayout, "[Sim 5] Nilai Komplementer =
"+ c.toString(), Snackbar.LENGTH_SHORT).show()
    }else if(a != 0 && b != 0 && c == 0 && i != 0 && s != 0){
        c = ( ( s - a ) - b) + i
        Snackbar.make(binding.rootLayout, "[Sim 11] Nilai Komplementer
= "+ c.toString(), Snackbar.LENGTH_SHORT).show()
    }
    }else if (NILAI == 6){ //i maksudnya irisan
        i = ( ( a + b ) + c ) - s
        Snackbar.make(binding.rootLayout, "[Sim 10] Nilai irisan = "+
i.toString() , Snackbar.LENGTH_SHORT).show()
    }else if (NILAI == 7){ //g maksudnya gabungan
        if (a != 0 && b != 0 && i != 0) {
            g = ( a + b ) - i
            Snackbar.make(binding.rootLayout, "[Sim 12] Nilai gabungan = "+
g.toString(), Snackbar.LENGTH_SHORT).show()
        }else{
            g = s - c
            Snackbar.make(binding.rootLayout,
"[Sim 13] Nilai gabungan = "+ g.toString(), Snackbar.LENGTH_SHORT).show()
        }
    }
}

```



Gambar 15. Aplikasi Penghitung Venn

KESIMPULAN

Berdasarkan hasil dan pembahasan didapati bahwa bahasa pemrograman Kotlin berbasis Android dapat dikombinasikan dengan bidang keilmuan matematika, dalam kasus ini dengan bahasan Teori Himpunan dengan bukti keberhasilannya melakukan simulasi yang diberikan.

DAFTAR PUSTAKA

Aminah, S., Wijaya, T. T., & Yuspriyati, D. (2018). Analisis Kemampuan Komunikasi Matematis Siswa Kelas VIII Pada Materi Himpunan. *Journal Cendekia: Jurnal Pendidikan Matematika*, 1(1), 15-22.

- Aulia, J., & Kartini. (2021). Analisis Kesalahan Siswa Dalam Menyelesaikan Soal Matematika Pada Materi Himpunan Kelas VII SMP/MTs. *Jurnal Cendekia: Jurnal Pendidikan Matematika*, 5(1), 484-500.
- Darwanto, Dinata, K. B., & Junaidi. (2020). *Teori Himpunan*. Lampung: Universitas Muhammadiyah Kotabumi.
- Elviani, Dina, Sugiatno, Sayu, & Silvia. (2020). Kemampuan Komunikasi Matematis Dikaji Dari Self-esteem Siswa Pada Materi Himpunan. *AlphaEuclidEdu*, 1(1), 1-8.
- Fitriyah, S. L., & Haerudin. (2021). Analisis Kemampuan Pemecahan Masalah Matematis Siswa SMP Pada Materi Himpunan. *UNION: Jurnal Pendidikan Matematika*, 9(2), 147-162.
- Hanifah. (2020). Analisis Kemampuan Pemecahan Masalah Soal Cerita Himpunan Oleh Mahasiswa Pendidikan Matematika FKIP UNIB. *Jurnal Pembelajaran Dan Pengajaran Pendidikan Dasar*, 3(1), 43-52.
- Jemerov, D., & Isakova, S. (2017). *Kotlin in Action*. Simon and Schuster.
- Khotimah, T., & Hilyana, F. S. (2017). Aplikasi Kalkulator Fisika Pencerminan Berbasis Android. *SNATIF*, 4, 15-20.
- Khotimah, T., & Hilyana, F. S. (2019). Aplikasi Konversi Pada Besaran Fisika Kinematika Berbasis Android. *SIMETRIS*, 10(2), 445-452.
- Penerapan Model Pembelajaran Kooperatif Tipe TAI Untuk Meningkatkan Kemampuan Pemahaman Matematis Siswa Pada Materi Himpunan. (2021). *Jurnal Pendidikan Guru Matematika*, 1(3), 326-333.
- Purwoko, H. (2019). Pemanfaatan Middleware Layer untuk Multi Platform pada Mobile dan Desktop. *STRING*, 4(2), 169-176.
- Rahmawan, F., & Kurniawan, I. (2019). Integrasi Nilai Keimanan dalam Materi Himpunan pada Pembelajaran Matematika. *Diskusi Panel Nasional Pendidikan Matematika*, 371-176.
- Ratnasari, L., & Abbasi, A. G. (2018). Perancangan Aplikasi Kalkulator Penilaian Kategorisasi Data Berbasis Android. *Jurnal Ilmiah Informatika Komputer*, 23(2), 136-143.
- Riana, & Ibrahim, M. (2019). LKS Himpunan: Sebuah Pengembangan Matematika Integrasi. *Jurnal Teori dan Aplikasi Matematika*, 3(2), 162-167.
- Santoso, W. A., Purwoko, H., & Rusmardiana, A. (2021). Aplikasi Penjualan Hewan Peliharaan Pada Toko Animalshop Di Jakarta Selatan Berbasis Android. *Seminar Nasional Riset dan Inovasi Teknologi*, 1171-1177.
- Vandini, I. (2015). Peran Kepercayaan Diri Terhadap Prestasi Belajar Matematika Siswa. *Jurnal Formatif*, 5(3), 210-219.



Wibowo, E., & Pratiwi, D. D. (2018). Pengembangan Bahan Ajar Menggunakan Aplikasi Kvisoft Flipbook Maker Materi Himpunan. *Desimal: Jurnal Matematika*, 1(2), 147-156.