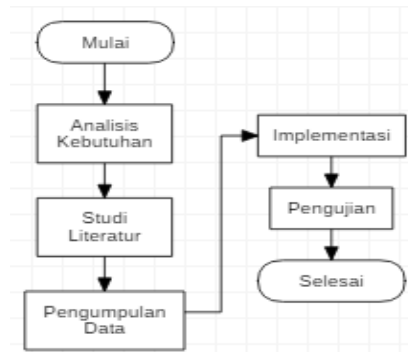


Metodologi

Tahapan Penelitian

Pengembangan sistem pencarian pada tujuh kitab hadis menggunakan algoritma *Knuth-Morris-Pratt* ini dilakukan dengan beberapa langkah yaitu analisis kebutuhan, studi literatur, pengumpulan data, implementasi, dan pengujian. Tahapan penelitian dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alur Penelitian.

a. Analisis Kebutuhan

Analisis kebutuhan dimulai dengan mengidentifikasi, mengumpulkan studi literatur mengenai metode-metode *string matching* khususnya algoritma *Knuth-Morris-Pratt*.

b. Studi Literatur

Studi literatur yang dilakukan pada penelitian ini berupa pengumpulan data yang dibutuhkan oleh sistem. Data-data yang dibutuhkan tersebut, diperoleh dari hasil melakukan pengamatan serta mempelajari catatan yang memuat data-data yang diperlukan terhadap permasalahan yang sedang diteliti. Metode studi literatur digunakan dengan melihat penelitian yang sudah ada dan merujuk pada penelitian yang telah dilakukan.

c. Pengumpulan Data

Data yang dibutuhkan pada penelitian ini yaitu data hadis yang diperoleh dari Al-Islam dan Hadist *Bot* yang dapat diakses menggunakan aplikasi Telegram.

d. Implementasi

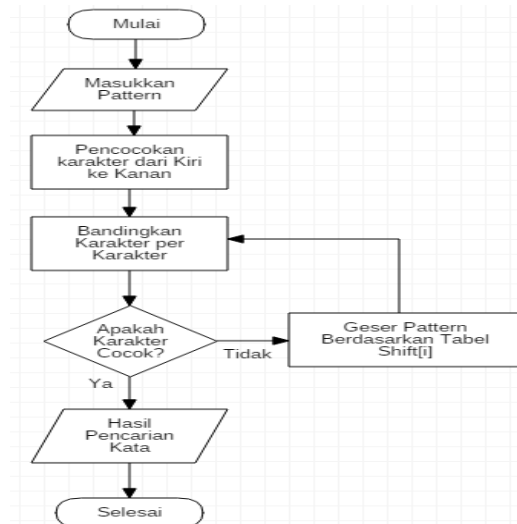
Implementasi merupakan tahapan peneliti melakukan analisis terhadap cara kerja dari algoritma *Knuth-Morris-Pratt* serta mengimplementasikannya ke dalam sebuah program dengan menggunakan PHP dan MySQL dalam melakukan proses pencarian kata.

e. Pengujian

Tahap ini merupakan tahap pengujian dari keseluruhan tahap-tahap yang telah dilalui dimulai dari analisis kebutuhan hingga tahap implementasi. Algoritma diuji dengan melakukan proses pencarian kata pada data hadis dalam sistem.

Cara Kerja Algoritma Knuth-Morris-Pratt

Algoritma *Knuth-Morris-Pratt* memiliki dua tahap pada proses pencariannya, yaitu tahap *pre-KMP* yang merupakan tahap pemberian nilai pergeseran pada masing-masing karakter yang dimasukkan (*pattern*) oleh *user*. Tahap kedua yaitu *KMPSearch* yang merupakan tahap pencocokkan karakter yang dimasukkan (*pattern*) oleh *user* dengan karakter yang ada pada database (*text*) berdasarkan nilai pergeseran. *Flowchart* dari pencarian menggunakan algoritma *Knuth-Morris-Pratt* dapat ditunjukkan pada Gambar 2.



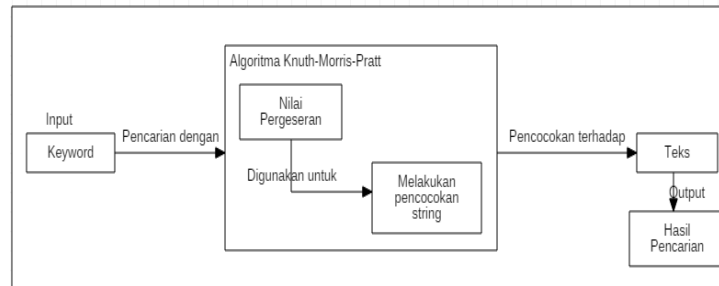
Gambar 2. *Flowchart* Algoritma *Knuth-Morris-Pratt*.

Gambar 2 menjelaskan alur pada proses pencarian algoritma *Knuth-Morris-Pratt*, dimana proses awal yang dilakukan yaitu memasukkan *pattern* di dalam *text* yang tersedia. Proses pencarian dimulai dari karakter paling kiri ke kanan *pattern*, setiap karakter akan dibandingkan satu per satu, jika terjadi ketidakcocokkan maka akan dicek nilai $shift[i]$ dan bergeser sesuai dengan nilai $shift[i]$, dan jika terjadi kecocokkan maka mengeluarkan hasil pencarian dan selesai.

3. Hasil dan Pembahasan

3.1 Hasil Penelitian

Hasil dari implementasi algoritma *Knuth-Morris-Pratt* dalam proses pencarian hadis yaitu sistem pencarian pada tujuh kitab hadis menggunakan algoritma *Knuth-Morris-Pratt*. Sistem pencarian hadis adalah sebuah sistem yang memiliki tujuan untuk mempermudah pencarian hadis, mempercepat waktu pencarian hadis dan dapat memberikan informasi rinci terkait hadis sesuai spesifikasi yang diinginkan oleh para pengguna yang membutuhkan informasi mengenai hadis. Penelitian dilakukan untuk mengimplementasikan algoritma *Knuth-Morris-Pratt* untuk melakukan pencarian. Proses pencarian dapat dilihat pada Gambar 3.

Gambar 3. Proses Pencarian Menggunakan Algoritma *Knuth-Morris-Pratt*.

Gambar 3 menggambarkan proses pencarian menggunakan algoritma *Knuth-Morris-Pratt*. Algoritma ini membagi proses pencarian dengan dua tahap yaitu proses mencari nilai pergeseran dari masing-masing karakter pada *keyword* yang dimasukkan *user*, dan proses pencocokkan karakter *keyword* yang telah memiliki nilai pergeseran pada teks yang tersedia dalam *database*.

3.2 Tahap *Pre-KMP*

Tahap *Pre-KMP* (*Pre-Knuth-Morris-Pratt*) merupakan suatu tahap pemberian nilai pergeseran terhadap *keyword* yang dimasukkan oleh *user*. Pemberian nilai ini dilakukan sebelum algoritma memulai pencocokkan *string* kedalam *database*, *keyword* yang dimasukkan akan dipecah menjadi *array* karakter. Tahap *pre-KMP* ini dapat mencegah pergeseran yang tidak berguna, proses *pre-KMP* hanya bergantung pada karakter yang ada di dalam *pattern*, tidak bergantung kepada karakter di dalam teks. Proses pemberian nilai pada proses *pre-KMP* dapat dilihat pada Gambar 4.

```

function preKMP($keyword) {
    $i = 0;
    $j = $kmpNext[0] = -1;
    while ($i < count($keyword)) {
        while ($j > -1 && $keyword[$i] != $keyword[$j]) {
            $j = $kmpNext[$j];
        }
        $i++;
        $j++;
        if (isset($keyword[$i]) == isset($keyword[$j])) {
            $kmpNext[$i] = $kmpNext[$j];
        } else {
            $kmpNext[$i] = $j;
        }
    }
    return $kmpNext;
}

$i++;
$j++;
if (isset($keyword[$i]) == isset($keyword[$j])) {
    $kmpNext[$i] = $kmpNext[$j];
} else {
    $kmpNext[$i] = $j;
}
return $kmpNext;
}
  
```

Gambar 4. Tahap *pre-KMP*.

Tahap *pre-KMP* ini menghasilkan nilai *kmpNext* untuk masing-masing karakter pada *array keyword* yang dimasukkan oleh *user*. Nilai *kmpNext* tersebut selanjutnya akan digunakan sebagai nilai lompatan pergeseran pada tahap selanjutnya yaitu tahap *KMP-Search*.

3.3 Tahap KMP-Search

Tahap kedua setelah tahap *pre-KMP* pada algoritma *Knuth-Morris-Pratt* adalah tahap *KMP-Search*. Tahap ini merupakan proses pencocokkan *string* yang dimasukkan *user* yang telah dipecah menjadi *array* dan memiliki nilai pergeseran setiap karakternya, terhadap data hadis yang terdapat pada *database*. Proses pencarian pada tahap *KMP-Search* dapat dilihat pada Gambar 5.

```
function KMPSearch($cari,$teks){
    $hasil = array();
    $cari_huruf_kecil= strtolower($cari);
    $teks_huruf_kecil= strtolower($teks);
    $pisah_keyword= str_split($cari_huruf_kecil);
    $pisah_teks= str_split($teks_huruf_kecil);
    $kmpNext= static::preKMP($pisah_keyword);
    $i = $j = 0;
    $num=0;
    while($j<count($pisah_teks)){
        while($i>=0&& $pisah_keyword[$i]!=$pisah_teks[$j]){
            $i = $kmpNext[$i];
        }
        $i++;
        $j++;
        if($i>=count($pisah_keyword)){
            $hasil[$num++]=$j-count($pisah_keyword);
            $i = $kmpNext[$i];
        }
    }
    return $hasil;
}
```

Gambar 5. Tahap KMP-Search.

Fungsi KMP yang telah dibuat kemudian dipanggil melalui *Controller* untuk melakukan pencarian seperti pada Gambar 6.

```
$Isi_Indonesia = $this->kmp-> KMPSearch($cari,$value->
    Isi_Indonesia);
```

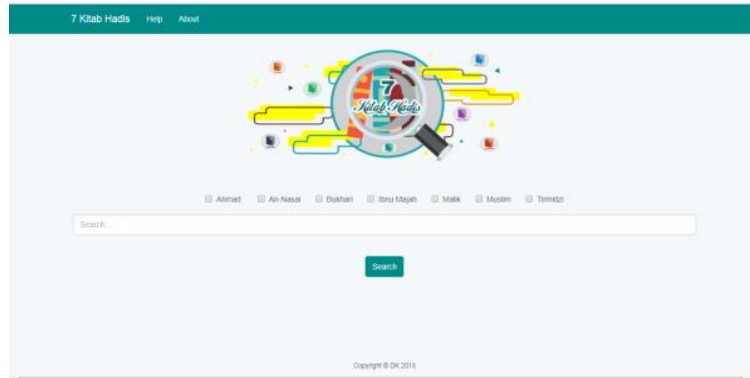
Gambar 6. Pemanggilan Fungsi KMP-Search.

Gambar 6 menunjukkan pemanggilan fungsi *KMP-Search* pada variabel *Isi_Indonesia* untuk melakukan pencocokkan karakter antara *keyword* yang dimasukkan *user* yang ditampung pada variabel *cari* dengan data pada *database* yang ditampung pada variabel *Isi_Indonesia*.

Tahap ini digunakan untuk mendapatkan hasil pencarian dari *keyword* yang dimasukkan *user* terhadap data yang ada pada *database*. Data hasil pencarian kemudian ditampilkan kepada *user* sebagai hasil pencarian yang ditemukan.

3.4 Tampilan Form Pencarian

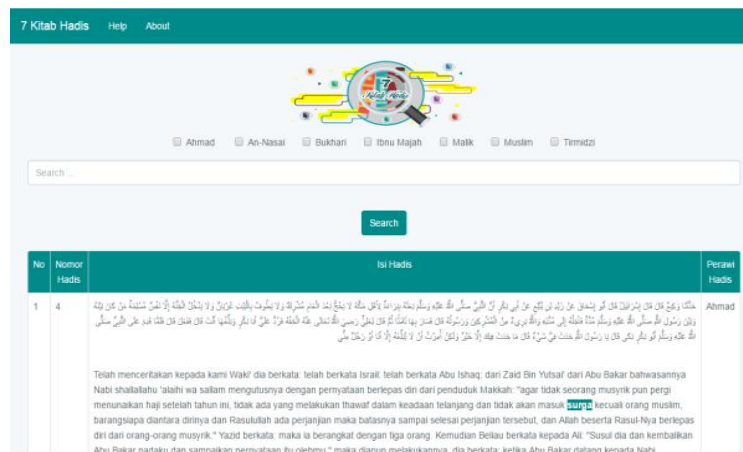
Tampilan pada sistem pencarian hadis pada tujuh kitab hadis ini dibuat dengan menampilkan *form* pencarian dengan pilihan kitab berupa pilihan *checkbox*, sehingga *user* bebas untuk memilih jumlah kitab yang akan dicari. *keyword* dimasukkan setelah memilih kitab yang akan dicari pada *form* yang telah disediakan. *Form* pencarian dapat dilihat pada Gambar 7.



Gambar 7. Form Pencarian.

3.5 Tampilan Hasil Pencarian

Tampilan hasil pencarian pada sistem pencarian hadis pada tujuh kitab hadis ini dibuat dengan menampilkan hasil pencarian dengan keterangan kitab, sehingga *user* dapat mengetahui perawi dari hadis yang ditampilkan. *Form* hasil pencarian pada kata “surga” dapat dilihat pada Gambar 8.



Gambar 8 Tampilan Hasil Pencarian.

3.6 Pengujian

Pengujian sistem yang dilakukan pada penelitian ini yaitu menguji apakah data dapat ditemukan pada saat proses pencarian menggunakan algoritma *Knuth-Morris-Pratt*, kemudian akan dibandingkan antara jumlah data yang ditemukan saat pencarian dengan algoritma dengan jumlah data yang ditemukan saat menggunakan *query* pada *database*. Hasil pengujian pencarian menggunakan *query* pada *database* dan pencarian menggunakan sistem dapat dilihat pada Tabel 1, Tabel 2, dan Tabel 3.

TABEL 1 Hasil Pencarian Dan Rata-Rata Waktu Pencarian Satu Kata

No	Kata yang dicari	Jumlah ditemukan (Tujuh kitab)	Rata-rata waktu Pencarian (Detik)
1	Puasa	2181	0.066000692
2	Haji	1933	0.065333683
3	Zakat	794	0.066967628
4	Surga	2524	0.066000619
5	Amal	1762	0.065666968

Tabel 1 menunjukkan hasil pencarian yang diperoleh dengan kata yang dicari yaitu Puasa, Haji, Zakat, Surga, dan amal. Kelima kata tersebut dicari sebanyak sembilan kali untuk mendapatkan rata-rata waktu pencarian.

Pencarian pada tujuh kitab hadis menghasilkan 2181 hadis yang mengandung kata “Puasa” dengan rata-rata waktu pencarian 0.066000692 detik, 1933 hadis yang mengandung kata “Haji” dengan rata-rata waktu pencarian 0.065333683 detik, 794 hadis yang mengandung kata “Zakat” dengan rata-rata waktu pencarian 0.066967628 detik, 2524 hadis yang mengandung kata “Surga” dengan rata-rata waktu pencarian 0.066000619 detik, dan 1762 hadis yang mengandung kata “Amal” dengan rata-rata waktu pencarian 0.065666968 detik.

TABEL 2 Hasil Pencarian Dan Rata-Rata Waktu Pencarian Dua Kata

No	Kata yang dicari	Jumlah ditemukan (Tujuh kitab)	Rata-rata waktu Pencarian (Detik)
1	Matahari Terbenam	172	0.076333690
2	Puasa Sunnah	14	0.074667632
3	Terbit Fajar	68	0.075000664
4	Hari Raya	256	0.074334326
5	Membayar Zakat	74	0.075200642

Tabel 2 menunjukkan hasil pencarian yang diperoleh dengan kata yang dicari yaitu Matahari Terbenam, Puasa Sunnah, Terbit Fajar, Hari Raya, dan Membayar Zakat. Kelima kata tersebut dicari sebanyak sembilan kali untuk mendapatkan rata-rata waktu pencarian.

Pencarian pada tujuh kitab hadis menghasilkan 172 hadis yang mengandung kata “Matahari Terbenam” dengan rata-rata waktu pencarian 0.076333690 detik, 14 hadis yang mengandung kata “Puasa Sunnah” dengan rata-rata waktu pencarian 0.074667632 detik, 68 hadis yang mengandung kata “Terbit Fajar” dengan rata-rata waktu pencarian 0.075000664 detik, 256 hadis yang mengandung kata “Hari Raya” dengan rata-rata waktu pencarian 0.074334326 detik, dan 74 hadis yang mengandung kata “Membayar Zakat” dengan rata-rata waktu pencarian 0.075200642 detik.

TABEL 3 Hasil Pencarian Dan Rata-Rata Waktu Pencarian Tiga Kata

No	Kata yang dicari	Jumlah ditemukan (Tujuh kitab)	Rata-rata waktu Pencarian (Detik)
1	Hanya Kepada Allah	16	0.079334264
2	Sebelum Matahari Terbenam	18	0.085333656
3	Dalam Api Neraka	16	0.075667213
4	Seorang Budak Perempuan	46	0.085000682
5	Bekas Perjalanan Jauh	1	0.084667452

Tabel 3 menunjukkan hasil pencarian yang diperoleh dengan kata yang dicari yaitu Hanya Kepada Allah, Sebelum Matahari Terbenam, Dalam Api Neraka, Seorang Budak Perempuan, Bekas Perjalanan Jauh. Kelima kata tersebut dicari sebanyak sembilan kali untuk mendapatkan rata-rata waktu pencarian.

Pencarian pada tujuh kitab hadis menghasilkan 16 hadis yang mengandung kata “Hanya Kepada Allah” dengan rata-rata waktu pencarian 0.079334264 detik, 18 hadis yang mengandung kata “Sebelum Matahari Terbenam” dengan rata-rata waktu pencarian 0.085333656 detik, 16 hadis yang mengandung kata “Dalam Api Neraka” dengan rata-rata waktu pencarian 0.075667213 detik, 46 hadis yang mengandung kata “Seorang Budak Perempuan” dengan rata-rata waktu pencarian 0.085000682 detik, dan satu hadis yang mengandung kata “Bekas Perjalanan Jauh” dengan rata-rata waktu pencarian 0.084667452 detik.

Rata-rata waktu pencarian dipengaruhi oleh panjang teks dan *keyword (pattern)* yang dicari oleh *user*. Semakin panjang *keyword (pattern)* dan teks pada *database* maka waktu pencarian akan meningkat, semakin sedikit *keyword (pattern)* dan teks pada *database* maka waktu pencarian akan semakin sedikit atau semakin cepat.

4. Simpulan

Simpulan yang diperoleh berdasarkan hasil pengujian dan analisis pada sistem pencarian pada tujuh kitab hadis menggunakan algoritma *Knuth-Morris-Pratt* adalah sebagai berikut:

1. Sistem pencarian pada tujuh kitab hadis menggunakan algoritma *Knuth-Morris-Pratt* berguna untuk menemukan hadis sesuai dengan yang dicari berdasarkan perawinya.
2. Algoritma *Knuth-Morris-Pratt* dapat diimplementasikan pada proses pencarian hadis.
3. Data yang disediakan sistem terdiri dari tujuh kitab hadis yaitu:
 - a. Kitab Ahmad dengan jumlah data 26.363 hadis.
 - b. Kitab An-nasa'i dengan jumlah data 5.662 hadis.
 - c. Kitab Bukhari dengan jumlah data 7.008 hadis.
 - d. Kitab Ibnu Majah dengan jumlah data 4.332 hadis.
 - e. Kitab Malik dengan jumlah data 1.594 hadis.
 - f. Kitab Muslim dengan jumlah data 5.362 hadis.
 - g. Kitab Tirmidzi dengan jumlah data 3.891 hadis.Dengan total jumlah hadis yaitu 52.618 hadis.
4. Panjang karakter teks pada *database* dan *pattern* yang dimasukkan oleh *user* mempengaruhi waktu pencarian, semakin panjang teks pada *database* dan *pattern* yang dimasukkan oleh *user* maka waktu pencarian yang dibutuhkan akan semakin lama.

Referensi

- [1] Anam, Syamsu. *Analisa Kinerja Search Engine Dengan Menilai Precision And Recall Untuk Informasi Marketing Dan Advertising (Studi Kasus: Google, Yahoo, MSN, Ask)*. Skripsi. Universitas Narotama Surabaya. Surabaya (2008)
- [2] Charras, C. Lecroq, T. *Handbook of Exact String-Matching Algorithms*. London. King's College Publications (2004)
- [3] Cooperman, Alan. Brian, J, Grim. Mehtab, S, Karim. Sahar, Chaudhry. *Mapping The Muslim Global Muslim Population*. Washington D.C. Pew Forum on Religion & Public Life (2009)
- [4] Ervana, A. dan Pertiwi, A. Implementasi Algoritma Pencocokan String pada Aplikasi Pengarsipan Berbasis Web. *Jurnal Informatika*. vol. 3, pp. 3-4 (2012)
- [5] Wibowo, Thio. Wibowo, Ardianto. Sari, Rika Perdana. Pembuatan Aplikasi Untuk Mendeteksi Kebenaran Perintah Sql Query Menggunakan Metode Knuth-Morris Pratt (KMP). *Jurnal Teknik Informatika*. vol. 1. pp. 3-5 (2012)

PENGEMBANGAN APLIKASI INFORMASI RUTE ANGKUTAN KOTA BANDAR LAMPUNG BERBASIS ANDROID

¹Rico Andrian, ²Febi Eka Febriansyah, ³Basir Efendi,

^{1,2,3}Jurusan Ilmu Komputer FMIPA Unila

¹rico.andrian@fmipa.unila.ac.id,

²febi.febrisyah@fmipa.unila.com, ³basir.efendi@gmail.com

Abstract

This research is done to design and build Information Application of Bandar Lampung City's Public Transportation with Android. This application can be used as a media to know about Bandar Lampung's Public Transportation and the routes which is used by Bandar Lampung City's public transportation, and the benefit of this application is to help citizen who does not know Bandar Lampung City to know about the routes of Bandar Lampung City's public transportation. Application of Bandar Lampung City's Public Transportation uses java programming language. Development of the application uses Unified Process (UP) method and is designed with Unified Modeling Language (UML). The result produced in the test using Equivalence Partitioning shows that system's management of rules are able to run well and the system can provide information well. Base on testing data, this application is categorized as User Friendly with the average of 83,67% (Very Good).

Keyword: Android, Angkot, Bandar Lampung City's Public Transportation

1. Pendahuluan

Angkutan umum adalah salah satu jenis sarana transportasi yang paling banyak digunakan oleh masyarakat sehari-hari, karena biaya yang relatif murah dan terjangkau oleh sebagian besar kalangan masyarakat. Kota Bandar Lampung adalah salah satu dari kota terbesar ketiga di pulau Sumatera yang mempunyai angkutan umum dengan populasi yang padat. Permasalahan yang perlu diperhatikan dalam transportasi angkutan umum di kota Bandar Lampung itu adalah kebingungan masyarakat dalam mencari jalur yang dilalui angkutan umum karena kurangnya informasi mengenai rute angkutan umum yang begitu banyak di kota Bandar Lampung, khususnya bagi masyarakat yang berkunjung ke kota Bandar Lampung. Pentingnya perencanaan rute dalam transportasi maka kebutuhan akan informasi mengenai rute angkutan umum di kota Bandar Lampung. [1]

Penelitian sebelumnya yaitu layanan berbasis Android yang diteliti oleh Singhal. Layanan ini, memiliki keuntungan bagi penggunanya, salah satunya yaitu untuk memperoleh informasi tentang lokasi pengguna saat ini. Layanan lokasi dapat diimplementasikan ponsel berbasis Android dengan memberi saran kepada pengguna kondisi lalu lintas saat ini, memberikan informasi rute, dan membantu menemukan hotel di sekitar lokasi. Metode yang digunakan dalam penelitian ini menggunakan LBS (*location based services*), untuk mengolah data lokasi di server dan meneruskan respon yang dihasilkan kepada pengguna, dan menemukan data lokasi untuk aplikasi berbasis perangkat *mobile* yang dapat menggunakan secara langsung. Hasil pada penelitian yang dilakukan, pengguna untuk mencari lokasi suatu tempat, menggunakan *google Places* API untuk mencari tempat beserta rincian mengenai lokasi.[2]

Penelitian sebelumnya yaitu sistem *real-time trip-planner* Transportasi Umum yang diteliti oleh Alves, memiliki kemampuan dalam mengetahui tentang rute terbaik. Proses *data mining* untuk mengalisis informasi perjalanan dan kecepatan, bertujuan untuk simulasi perjalanan. Metode yang