



SIMPEL (Sistem Informasi Manajemen Pelatihan) Internal BRI Menggunakan Metode *Agile* dengan Model *Extreme Programming* dan Algoritma *Brute Force*

Anang Kurniawan ^{1*}, Fauziah ²

^{1,2} Program Studi Sistem Informasi, Fakultas Teknologi Komunikasi dan Informatika, Universitas Nasional, Kota Jakarta Selatan, Daerah Khusus Ibukota Jakarta, Indonesia.

article info

Article history:

Received 11 August 2022

Received in revised form

8 December 2022

Accepted 16 February 2023

Available online April 2023

DOI:

<https://doi.org/10.35870/jtik.v7i2.754>

Keywords:

SIMPEL (Training Management Information System); BRI Internal; Agile Method; Extreme Programming Model; Brute Force Algorithm.

Kata Kunci:

SIMPEL (Sistem Informasi Manajemen Pelatihan); Internal BRI; Metode Agile; Model Extreme Programming; Algoritma Brute force.

abstract

This research develops SIMPEL (Training Management Information System) Internal BRI using Agile Methods, especially the Extreme Programming (XP) Method, and the Brute force Algorithm as a tool. XP is used because it is known as a method that is fast and flexible in accommodating changing user needs. The Brute force algorithm is implemented to help users search for training or similar information. White Box Testing is carried out to test the algorithm and shows that the time needed to find a participant data value is 0.43 ms. The results of this study indicate that the use of Agile Methods and Brute force Algorithms can increase efficiency and flexibility in software development.

abstract

Penelitian ini mengembangkan SIMPEL (Sistem Informasi Manajemen Pelatihan) Internal BRI dengan menggunakan Metode Agile, terutama Metode Extreme Programming (XP), dan Algoritma Brute force sebagai alat bantu. XP digunakan karena dikenal sebagai metode yang cepat dan fleksibel dalam mengakomodasi perubahan kebutuhan pengguna. Algoritma Brute force diimplementasikan untuk membantu pengguna mencari pelatihan atau informasi serupa. Pengujian White Box Testing dilakukan untuk menguji algoritma tersebut dan menunjukkan bahwa waktu yang dibutuhkan untuk mencari suatu nilai data peserta adalah 0,43 ms. Hasil dari penelitian ini menunjukkan bahwa penggunaan Metode Agile dan Algoritma Brute force dapat meningkatkan efisiensi dan fleksibilitas dalam pengembangan perangkat lunak.

Corresponding Author. Email: ridwan.sastrawijaya@gmail.com ^{1}.

1. Latar Belakang

PT Bank Rakyat Indonesia TBK, merupakan salah satu BUMN terbesar di Indonesia milik pemerintah yang bergerak di bidang perbankan, dan berfokus terhadap usaha mikro kecil menengah (UMKM). Visi BRI tahun 2025 adalah *The Most Valuable Banking Group In Southeast Asia & Champion of Financial Inclusion*, untuk mencapai visi BRI 2025 memerlukan Sumber Daya Manusia yang unggul dan berkualitas. Untuk mempertahankan dan meningkatkan keunggulan Sumber Daya Manusia bukanlah hal yang mudah. Perlu adanya pelatihan secara berkesinambungan. Sumber Daya Manusia yang unggul dan berkualitas tentu harus ditunjang dengan infrastruktur yang memadai. Oleh karena itu Penulis mencoba membuat infrastruktur yang berkualitas, khususnya dibidang pelatihan dan pendidikan. Seperti karyawan BRI dapat melakukan pelatihan pendidikan melalui online tanpa harus datang ketempat yang telah ditentukan. Tujuan dari pembuatan *website* SIMPEL (Sistem Informasi manajemen Pelatihan) ini adalah untuk mempermudah administrasi, efisiensi biaya pendidikan dan mengatasi permasalahan yang terjadi dimana ada beberapa case antara lain seperti peserta sedang mengikuti pelatihan A tetapi bisa mengikuti pelatihan B, tentu hal ini akan membuat peserta pendidikan tidak fokus dan maksimal untuk mengupdate skill Knowledge dengan adanya SIMPLE ini peserta tidak akan bisa lagi mengikuti double pelatihan agar peserta bisa lebih fokus untuk mengikuti pelatihan dan ilmu yang di dapatkan bisa di terapkan di diri sendiri atau unit kerjanya.

Pada aplikasi ini nantinya terdiri dari 2 aktor yaitu admin, dan juga peserta pelatihan. Yang dimana admin dan peserta berasal dari karyawan internal BRI. Untuk pemateri atau pengajar berasal dari eksternal BRI. Dan untuk pemateri tidak dapat mengakses *website* pelatihan ini. Jika pemateri ingin mengatur jadwal pelatihannya ataupun ingin memberikan soal. Pemateri dapat menghubungi admin. Yang dimana fungsi dari admin ini adalah untuk menambahkan peserta, melakukan pengeditan peserta, menghapus peserta dan juga pemateri. Admin juga dapat menambah, mengedit, ataupun menghapus, judul pelatihan. Dan juga dapat menambah, mengedit, menghapus *self-learning* atau modul. Selain itu admin juga dapat menambah,

mengedit, menghapus soal atau *quiz*. Dan juga admin dapat melihat seluruh peserta. Peserta mengikuti pelatihan pendidikan berdasarkan dari surat rekomendasi dari atasan, dan mengikuti pelatihan jika telah ditambahkan oleh admin. Peserta juga berhak untuk mengikuti ataupun menolak pelatihan yang telah ditentukan oleh admin dan atasan. Peserta mengikuti pelatihan bervariasi sesuai dengan jadwal yang telah ditentukan oleh admin. Dengan rentan waktu minimal 2 minggu sampai dengan 1 bulan. Peserta dinyatakan lulus jika peserta mengikuti pelatihan. Dengan mengikuti *selflearning* dan mengerjakan *quiz* sesuai dengan jadwal yang telah ditentukan.

Sistem terdiri dari 3 unsur utama yaitu input (masukan), *processing* (pengolahan) dan *output* (keluaran). Pengertian dari sistem adalah sebuah *instrument* yang saling terhubung dan bekerja sama guna mencapai suatu tujuan yang diinginkan [1]. Sekumpulan data yang telah diubah atau dikelompokkan dan diterjemahkan untuk dipakai di dalam proses pengambilan keputusan merupakan pengertian dari sebuah Informasi [2]. Seni menyelesaikan pekerjaan melalui orang lain merupakan pengertian manajemen dari Mary Parker Follet [3]. Pengertian dari Sistem informasi manajemen adalah sistem informasi yang dipakai untuk menyuguhkan informasi yang diimplementasikan sebagai penunjang sistem manajemen dan pengambilan keputusan dalam suatu lembaga, badan ataupun organisasi [4]. mPDF adalah sebuah library php yang membuat sebuah file berupa file PDF dalam bentuk UTF-8 encodedHTML. Pembuatan mPDF ini didasarkan pada FPDF dan juga HTML 2 FPDF dengan sedikit perubahan yang membuat mPDF menjadi lebih konsisten. Penulis asli dari mPDF adalah lan Back, dia mengeluarkan filePDF 'on-the-fly' dari situs webnya yang dapat menangani bahasa web yang berbeda. Selain itu, mPDF dapat menggunakan CSS. [5].

2. Metode Penelitian

Di dalam membuat aplikasi ini penulis membuat 3 metode sebagai landasannya yang pertama Metode *Agile*, Metode *Extreme Programming*, dan yang terakhir adalah Algoritma *Brute force*. Metode *Agile* digunakan untuk membuat aplikasi secara cepat, fleksibel tidak

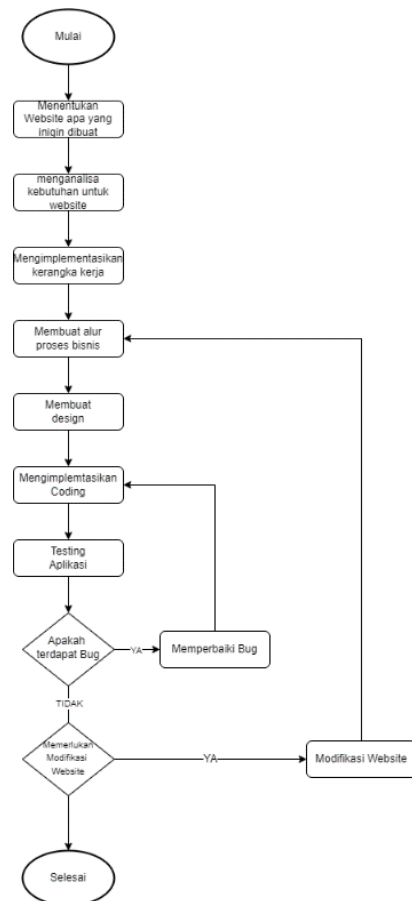
takut dengan perubahan yang terjadi secara mendadak berbeda dengan metode yang lama (*waterfall*). Metode *Agile* biasa juga dikenal dengan nama framework *Agile* dan mempunyai produk turunan salah satunya adalah Metode *Extreme Programming* dalam penerapannya hanya menggunakan 4 fase. Fase *planning*, *Design*, *Coding*, *Testing*. Lebih sedikit 2-3 dari produk turunan *Agile* lainnya. Yang Diama biasanya memiliki 6-7 fase. Dan Terakhir ada algoritma *Brute force* yang memudahkan admin nanti dalam pencarian data nantinya.

Metode Agile

Dalam Membuat sebuah *website* banyak cara yang dapat digunakan dengan berbagai macam metode yang ada. Seperti metode *iterative*, metode *big bang*, metode *prototype*, metode UP, metode RAD, metode *spiral*, metode *Agile*, metode *waterfall*, *v-model*, metode *scrum*, metode RUD, metode fountain dan lain - lain [6]. Setelah melihat banyaknya metode yang ada. Penulis memutuskan untuk menggunakan metode *Agile* karena sesuai dengan rancangan *website* yang akan penulis buat. Metode *Agile* sendiri menjadi alternatif yang digunakan selain metode *waterfall* yang dimana metode tersebut membutuhkan waktu yang lama. Metode *Agile* berbeda dengan metode *sdic* lama. Metode *Agile* adalah metodologi pembentukan perangkat lunak yang menggunakan dasar - dasar sistem jangka pendek yang diharuskan menyesuaikan secara cepat terhadap perubahan yang terjadi dalam bentuk apapun request dari client. [7]. Metode *Agile* biasa juga disebut dengan framework *Agile*. Macam - Macam dari metode *Agile* seperti *Scrum*, Metode *Crystal*, *Extreme Programming*, *SAFe*, *Kanban*, *LSD*, dan lain - lain. Diantara macam - macam metode *Agile* tersebut penulis memutuskan untuk menggunakan Metode *Extreme Programming*.

Metode *Extreme Programming* (XP) merupakan salah satu dari framework *Agile* yang digunakan dalam pembuatan aplikasi ini. XP fokus pada pengembangan perangkat lunak yang cepat, berkualitas tinggi, dan responsif terhadap perubahan yang terjadi [8]. Dalam XP, pengembangan dilakukan dengan cara melakukan serangkaian aktivitas yang berulang-ulang, seperti perencanaan, desain, pengkodean, dan pengujian, yang dilakukan secara bersamaan dan kolaboratif oleh tim yang terdiri dari programmer, pengguna, dan pelanggan. Tujuan dari XP adalah untuk memberikan solusi yang efektif dan

efisien untuk kebutuhan pelanggan, dengan menekankan pada komunikasi yang baik, disiplin teknis, dan umpan balik yang cepat. Penerapan XP dalam pembuatan aplikasi ini difokuskan pada empat fase yaitu *planning*, *design*, *coding*, dan *testing*. Selain metode *Agile* dan XP, penulis juga menggunakan algoritma *Brute force* dalam pembuatan aplikasi ini. Algoritma *Brute force* merupakan algoritma pencarian data dengan melakukan pengecekan secara bertahap pada setiap kemungkinan solusi, sehingga memudahkan admin dalam pencarian data pada aplikasi yang dibuat. Dengan kombinasi ketiga metode tersebut, diharapkan aplikasi yang dibuat dapat memiliki kualitas yang baik dan dapat memenuhi kebutuhan pelanggan.



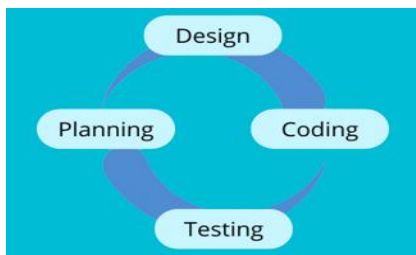
Gambar 1. Flowchart Pembuatan Website

Gambar diatas adalah gambar 1. Flowchart proses pembuatan *website* ini. Yang dimana dimulai dari menentukan *website* apa yang dibuat, Dilanjutkan dengan menganalisa kebutuhan *website*, mengimplementasikan kerangka kerja yang telah ditentukan, setelah itu membuat alur proses bisnis, membuat *design*, mengimplementasikan *coding*, testing

aplikasi, lalu ada pengecekan apakah terdapat sebuah bug atau tidak, jika iya maka harus diperbaiki dan dilakukan *coding* ulang, jika tidak maka akan lanjut ketahap berikutnya. Apakah *website* ini akan ada perubahan lagi atau tidak. Jika tidak maka proses pembuatan *website* sudah selesai. Tetapi jika iya maka akan dilakukan tahapan ulang dari alur proses bisnis hingga tahap testing aplikasi dan akhirnya selesai.

Metode Extreme Programming

Alasan penggunaan metode *Extreme Programming* dalam Metode *Agile* ini dikarenakan sifat dasar dari sistem ini yang sangat cepat hanya melalui 4 fase dari fase *planning*, fase *design*, fase *coding*, dan fase testing lebih sedikit dari 2-3 fase yang ada. [8]. Keempat fase tersebut terus menerus berulang secara berurutan dari *planning*, *design*, *coding*, testing dan Kembali ke *planning* kembali sampai dengan testing. Metode ini tidak kaku, metode ini fleksibel sangat menerima perubahan jika diperlukan. Nilai – nilai dasar *Extreme Programming* ada 5 yaitu: *communication*, *courage*, *simplicity*, *feedback*, *quality work* [9]. *Communication*: Komunikasi yang cair antara penulis dengan *user*, *courage*: Penulis harus memiliki keberanian, keyakinan serta integritas dalam menyelesaikan *website* ini, *Simplicity*: menerapkan semua tugas secara sesederhana mungkin, *Feedback*: Adanya feedback atau masukan dari user berguna mengembangkan *website* ini menjadi lebih baik lagi, dan yang terakhir *Quality Work*: Proses pengerjaan yang dilakukan dengan sungguh-sungguh menghasilkan aplikasi *website* sesuai dengan apa yang diinginkan.



Gambar 2. Proses *Extreme Programming*

Gambar diatas adalah gambar 2. Fase–Fase yang terjadi pada Metode *Extreme Programming* Yang dimulai dari fase *planning*, fase *design*, fase *coding* dan yang terakhir tetapi tidak berakhir adalah fase testing. Selanjutnya Fase tersebut akan mengulang lagi ke fase *planning*, *design*, *coding*. Terus berulang secara *iterative*.

1) Fase *Planning*

Pada Fase ini adalah fase dimana penulis

melakukan perencanaan dengan client. Yang dimana penulis membuat analisa proses bisnis. Mengumpulkan kebutuhan sistem, fitur-fitur fungsionalitas yang akan digunakan pada sistem perangkat lunak. Hasilnya adalah: Disepakati *Website* ini terdiri dari 2 Aktor yaitu Peserta, dan Admin. Untuk pengajar/pemateri karena berasal dari eksternal bank BRI maka tidak dibuatkan akun. Jika ada berkaitan dengan proses pelatihan, maka pengajar akan berkomunikasi dengan admin. Seperti penambahan modul self learning sebagai bahan belajar peserta. Dan juga ujian yang akan dikerjakan oleh peserta apakah peserta memahami yang telah diberikan atau tidak.

2) Fase *Design*

Setelah melakukan fase *planning*, fase selanjutnya adalah fase *design*. Yang dimana penulis membuat Rancangan sistem lanjutan dari fase sebelumnya, hasil dari fase ini adalah UML (Undifined Modeling Language). Dan setelah itu ada pembuatan *design website* untuk UI (User Interface). Dan Juga Rancangan Database.

3) Fase *Coding*

Fase *coding* adalah fase ketiga setelah fase *planning* dan fase *design*. Fase ini mengimplementasikan *coding* mengikuti aturan dasar yang telah disepakati agar memudahkan penulis. dan tetap pada jalur yang telah disepakati. Di database ini menggunakan mysql, backendnya menggunakan php7, dan frontendnya menggunakan bootstrap 5.

4) Fase *Testing*

Fase ini adalah fase terakhir setelah melakukan pengcodingan yaitu fase testing. Fase ini biasa disebut juga fase pengujian. Fase pengujian, disini perangkat lunak yang sudah melewati fase *coding* akan diuji. Apakah code tersebut sudah sesuai dengan apa yang diinginkan atau terdapat bug didalamnya. jika terdapat bug didalamnya maka harus diperbaiki dan diuji kembali sampai tidak terdapat bug didalamnya. Penulis menggunakan White Box Testing sebagai bahan uji aplikasi. Penulis menguji logika pencarian algoritma *brute force* ini yang dimana berapa waktu yang dihabiskan untuk mencari nilai pada suatu data yang ada di dalam *website* ini.

Algoritma *Brute force*

Rangkaian dari separuh atau seluruh tindakan yang logis untuk memberikan solusi adalah pengertian dari algoritma menurut [10]. Algoritma yang

diimplementasikan untuk mencocokkan dengan semua string atau pattern antara 0 dan n-m untuk mendapatkan keberadaan Pattern string yang sesuai merupakan pengertian dari algoritma *brute force* [11].

3. Hasil dan Pembahasan

UML (Unified Modeling Language)

UML (Unified Modeling Language) adalah suatu metode dalam pemodelan secara visual yang diterapkan sebagai landasan perencanaan sistem berorientasi objek. UML ini memiliki sejumlah produk turunan seperti Sequence Diagram, Component Diagram, Use Case Diagram, State Machine Diagram, Activity Diagram dan Class Diagram. Disini Penulis memutuskan untuk menggunakan Use Case Diagram sebagai salah satu bentuk turunan dari UML. Use Case Diagram mampu menguraikan tipe hubungan antara sipengguna sistem dengan sistemnya.



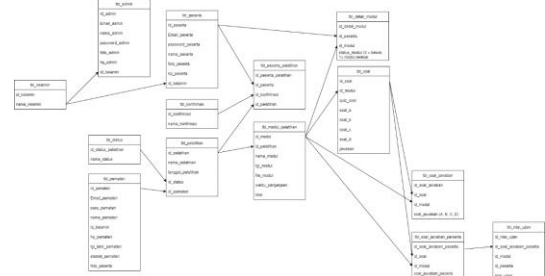
Gambar 3. Use Case Diagram

Gambar diatas adalah gambar 3. Gambar Use Case Diagram. Gambar tersebut mempunyai arti sistem terdiri dari 2 aktor yaitu Admin dan Peserta. Admin dapat melakukan Mengoperasikan Data Admin, Mengoperasikan Data Peserta, Mengoperasikan Data Pengajar, mengoperasikan data pelatihan, mengoperasikan data peserta pelatihan, mengoperasikan data jadwal pelatihan, mengoperasikan data self learning, mengoperasikan data ujian. Semua aktivitas admin harus melakukan validasi login terlebih dahulu. Jika tidak maka aktivitas aktor admin tidak dapat Dilakukan. Setelah itu ada aktor peserta yang dimana aktor peserta dapat melakukan mengikuti pelatihan sesuai jadwal, mengikuti self learning, mendapatkan niali ujian,

mengerjakan ujian Dan semua aktivitas peserta harus melakukan validasi login terlebih dahulu. Jika tidak maka aktivitas aktor peserta tidak dapat dilakukan.

Relasi Database

Tabel relasi diimplementasikan untuk membagi data menjadi tabel-tabel yang menetapkan entitas dari relasi yang bertindak untuk mengiimplementasikan data sehingga database tersebut dapat mudah diubah. Berikut ini digambarkan relasi antar tabel :



Gambar 4. Relasi Database

Gambar diatas adalah gambar 4. Relasi Database. Terdapat 14 tabel. Yang dimana 14 tabel tersebut adalah tbl_admin, tbl_peserta, tbl_kelamin, tbl_status, tbl_pemateri, tbl_konfirmasi, tbl_pelatihan, tbl_peserta_pelatihan, tbl_modul_pelatihan, tbl_detail_modul, tbl_soal, tbl_soal_jawaban, tbl_soal_jawaban_peserta, tbl_nilai_ujian.. Yang dimana masing- masing table tersebut saling berelasi

Whitebox Testing

White Box Testing mempunyai beberapa nama lain seperti transparent box testing, clear box testing, glass box testing, open box testing, structural testing. Yang intinya adalah sebuah pengujian aplikasi yang dimana berfungsi untuk menemukan sebuah bug yang ada diaplikasi yang penulis buat, merupakan pengertian dari White Box Testing. Sebuah aplikasi atau *website* dinyatakan berhasil apabila sudah sesuai dengan apa yang diharapkan dan tidak terdapat bug didalamnya. Disini penulis akan menguji sebuah *codingan* dengan algoritma *brute force*.

Algoritma Brute force

Algoritma untuk memberikan solusi menggunakan perencanaan secara gamblang, proposional, dan lugas yang diimplementasikan dalam menyamakan string dengan menyelaraskan tiap string dengan string lainnya merupakan pengertian algoritma *brute force* [12][13]. Selanjutnya adalah proses algoritma *brute force* dalam mennyamakan string.

Contoh-contoh

(Berdasarkan definisi konsep yang terlibat)

1. Menghitung a^n ($a > 0$, n adalah bilangan bulat tak-negatif)

Definisi:

$$a^n = a \times a \times \dots \times a \quad (n \text{ kali}), \text{ jika } n > 0$$

$$= 1, \text{ jika } n = 0$$

Algoritma *brute force*: kalikan 1 dengan a sebanyak n kali

Gambar 5. Rumus Algoritma *Brute force*

Gambar diatas adalah gambar 5. rumus dalam menggunakan algoritma *brute force*. Diketahui an adalah mencari kata atau data yang sama. Dan a adalah algoritma *brute force* yang mencari kata yang sama atau pattern yang sama. Jika n kali melebihi dari kata atau data yang dicari maka hasilnya adalah nol atau false maka nilai n kali akan terus mencari pattern yang sama untuk menghasilkan nilai true yang nantinya akan menampilkan data di tampilan layar penulis.

```

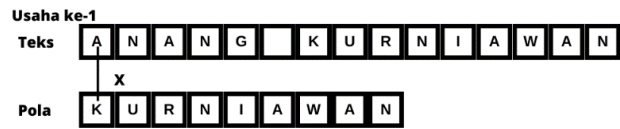
1 <?php
2 $pecah;
3 $n = strlen($pecah['nama_peserta']) OR strlen($pecah['id_peserta']);
4 $m = $keyword;
5
6 function bruteForce(){
7     $n = 0;
8     $m = 0;
9
10    for ($i=0; $i <= $n - $m ; $i++) {
11        $j = 0;
12        while($j < $m && $pecah[$i + $j] == $keyword[$j]){
13            $j++;
14        }
15        if ($j == $m) return $i;
16    }
17    return -1;
18 }
19
20 $hasil =microtime(bruteForce());
21 $nilai = substr($hasil, 0 , 6);
22 $num = (double)$nilai;
23 echo "<p class='fw-bold'>waktu pencarian : ". $num ;
24 >

```

Gambar 6. Implementasi Algoritma *Brute force*

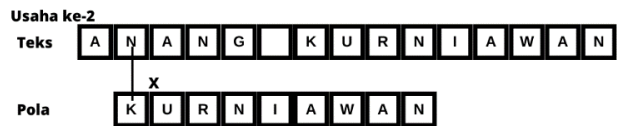
Gambar diatas adalah gambar 6. hasil implementasi dari rumus algoritma *brute force* pada gambar 5. Yang sebelumnya. Jika dilihat dengan seksama pada gambar diatas terdapat \$pecah adalah array dari peserta pelatihan. \$n adalah sebuah nilai nama peserta atau id peserta yang nilainya akan dicari menggunakan rumus algoritma *brute force*. \$m adalah inputan yang akan penulis masukkan guna mencari pattern yang sama. Yang mana semuanya akan dieksekusi disebuah *function bruteForce()*. Dibawah ini terdapat contoh bagaimana algoritma *brute force* bekerja dengan skema salah satu peserta dengan nama “Anang Kurniawan”. Dan penulis melakukan input data dengan nama “Kurniawan”. Berikut adalah gambaran bagaimana algoritma *brute force* ini bekerja menyelaraskan sebuah string guna mencari pattern atau pola untuk menghasilkan sebuah nilai

ture dan akhirnya menampilkan data ke layer minitor.



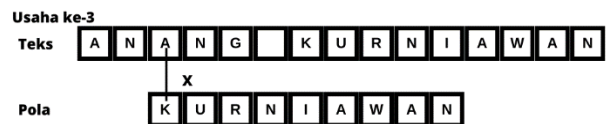
Gambar 7. Usaha ke-1

Gambar diatas adalah gambar 7. Diketahui terdapat nama peserta dengan teks “Anang Kurnianwan”, Penulis melakukan input data dengan pola nama “Kurniawan”. Usaha pertama diketahui Huruf pertama pada teks “Anang Kuniawan” adalah “A” sedangkan Huruf pertama pada pola “Kurniawan” adalah “K” maka hasilnya adalah tidak sama. Sehingga harus melakukan pada usaha kedua.



Gambar 8. Usaha Ke-2

Gambar diatas adalah gambar 8. Diketahui terdapat nama peserta dengan teks “Anang Kurnianwan”, Penulis melakukan input data dengan pola nama “Kurniawan”. Usaha kedua diketahui Huruf kedua pada teks “Anang Kuniawan” adalah “N” sedangkan Huruf pertama pada pola “Kurniawan” adalah “K” maka hasilnya adalah tidak sama. Sehingga harus melakukan pada usaha ketiga



Gambar 9. Usaha Ke-3

Gambar diatas adalah gambar 9. Diketahui terdapat nama peserta dengan teks “Anang Kurnianwan”, Penulis melakukan input data dengan pola nama “Kurniawan”. Usaha ketiga diketahui Huruf ketiga pada teks “Anang Kuniawan” adalah “A” sedangkan Huruf pertama pada pola “Kurniawan” adalah “K” maka hasilnya adalah tidak sama. Sehingga harus melakukan pada usaha keempat.



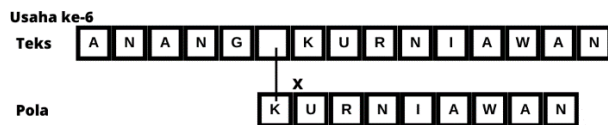
Gambar 10. Usaha Ke-4

Gambar diatas adalah gambar 10. Diketahui terdapat nama peserta dengan teks “Anang Kurnianwan”, Penulis melakukan input data dengan pola nama “Kurniawan”. Usaha keempat diketahui Huruf keempat pada teks “Anang Kuniawan” adalah “A” sedangkan Huruf pertama pada pola “Kurniawan” adalah “K” maka hasilnya adalah tiak sama. Sehingga harus melakukan pada usaha kelima.



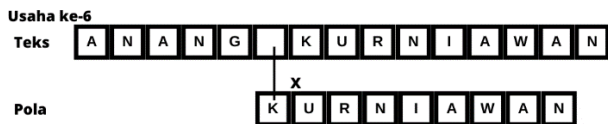
Gambar 11. Usaha Ke-5

Gambar diatas adalah gambar 11. Diketahui terdapat nama peserta dengan teks “Anang Kurnianwan”, Penulis melakukan input data dengan pola nama “Kurniawan”. Usaha kelima diketahui Huruf kelima pada teks “Anang Kuniawan” adalah “G” sedangkan Huruf pertama pada pola “Kurniawan” adalah “K” maka hasilnya adalah tidak sama. Sehingga harus melakukan pada usaha keenam



Gambar 12. Usaha Ke-6

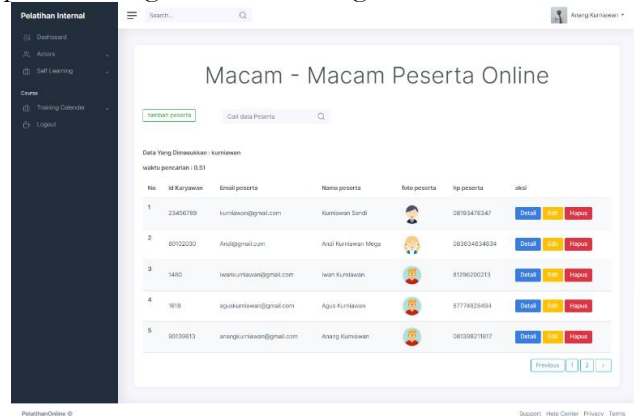
Gambar diatas adalah gambar 12. Diketahui terdapat nama peserta dengan teks “Anang Kurnianwan”, Penulis melakukan input data dengan pola nama “Kurniawan”. Usaha kedua diketahui Huruf kedua pada teks “Anang Kuniawan” adalah “ ” sedangkan Huruf pertama pada pola “Kurniawan” adalah “K” maka hasilnya adalah tidak sama. Sehingga harus melakukan pada usaha ketujuh.



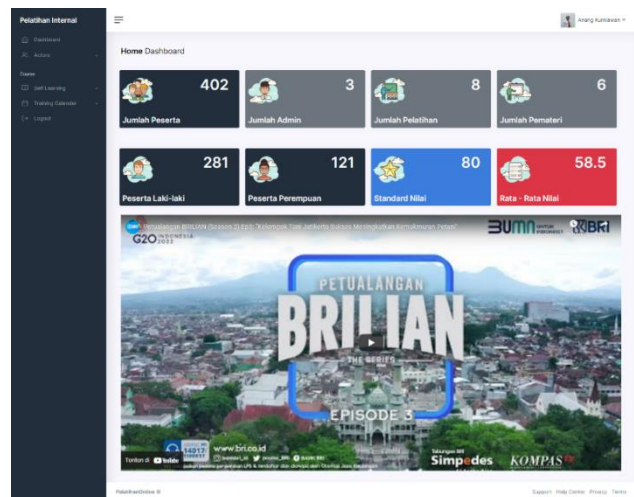
Gambar 13. Usaha Ke-7

Gambar diatas adalah gambar 13. Diketahui terdapat nama peserta dengan teks “Anang Kurnianwan”, Penulis melakukan input data dengan pola nama “Kurniawan”. Usaha ketujuh diketahui Huruf ketujuh pada teks “Anang Kuniawan” adalah “K” sedangkan Huruf pertama pada pola “Kurniawan”

adalah “K” maka hasilnya adalah benar. Lalu pada huruf kedelapan pada teks adalah “U” dan huruf kedua pada pola juga “U” maka hasilnya adalah benar. Setelah itu maka huruf teks dan pola akan saling menyelaraskan sampai dengan kedua teks dan pola mengashilkan nilai yang sama yaitu “Kurniawan”. Hasilnya adalah *website* akan menampilkan nama peserta dengan nama “Anang Kurniawan”.

Gambar 14. Hasil dari Algoritma *Brute force*

Gambar diatas adalah gambar 14. Hasil dari pencarian dengan pola nama “kuniawan”. Menampilkan nama peserta Anang Kurniawan, Agus Kurniawan, Iwan Kurniawan, Andi Kurniawan Mega, Kurniawan Sandi. Pencarian tersebut memakan waktu 0.3235 dan 5 orang peserta dari sebuah 402 peserta.



Gambar 15. Home Dashboard Admin

Gambar diatas adalah gambar 15. Tampilan Home Dashboard Admin Yang berisi Jumlah peserta 402, Jumlah admin 3, Jumlah Pelatihan 8, Jumlah pemateri 6. Dan jumlah tersebut dapat bertambah seiringnya waktu pertambahan peserta maupun pertambahan jumlah pelatihan dan jumlah pemateri. Dan

dibawahnya terdapat 4 card lainnya seperti informasi jumlah peserta yang berjenis kelamin laki-laki, jumlah peserta dengan kelamin perempuan, Standard nilai yang ditetapkan untuk setiap pelatihan. Dan terakhir terdapat rata-rata nilai pelatihan. Jika nilai kurang dari standard maka akan berwarna merah jika lebih dari sama dengan nilai standard maka hasilnya akan berwarna hijau.



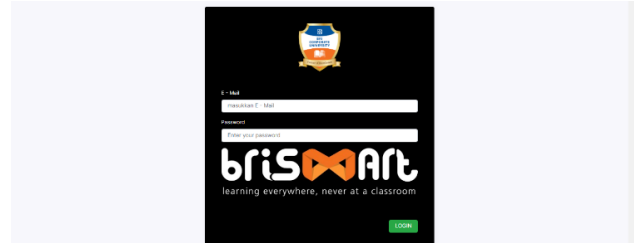
Gambar 16. Home Dashboard Peserta

Gambar diatas adalah gambar 16. Tampilan Home Dashboard Peserta Yang berisi Jumlah pelatihan yang sudah diambil oleh peserta, jumlah self learning yang sudah diambil, standard nilai yang telah ditetapkan oleh pihak BRI. Dan yang terakhir rata-rata nilai setiap mengerjakan self learning jika peserta dibawah rata-rata maka warnanya akan menjadi merah, dan jika diatas standard maka warnanya akan menjadi hijau.



Gambar 17. Sertifikat Pelatihan

Gambar diatas adalah gambar 17. Jika Peserta telah menyelesaikan pelatihan sesuai dengan standard nilai yang telah ditentukan. Maka peserta dinyatakan lulus dan mendapatkan sertifikat sebagai bukti bahwa peserta telah lulus menyelesaikan pelatihan.



Gambar 18. Halaman Menu Login

Gambar diatas adalah gambar 18. Tampilan halaman dari menu login, yang dimana baik peserta maupun admin harus melewati Langkah ini. Jika email ataupun password yang dimasukkan salah. Maka baik peserta maupun admin tidak dapat masuk ke halaman utama. Dan akan Kembali ke halaman ini untuk melakukan validasi login Kembali

Tabel 1. Hasil pencarian menggunakan algoritma *brute force* pada data peserta

No	Pattern	Jumlah peserta yang ditemukan	<i>Brute force</i> (ms)
1	Kurniawan	5	0,51
2	Anang	1	0,19
3	Budi	6	0,54
4	Agus	13	0,26
5	Agung	1	0,11
6	Muhammad	3	0,46
7	Ahmad	5	0,24
8	Ridwan	1	0,40
9	Nurul	1	0,84
10	Putri	5	0,75

Berdasarkan table diatas diketahui terdapat 10 pattern nama peserta terdiri dari Kurniawan, anang, budi sampai dengan yang terakhir adalah putri. Hasil yang didapatkan dari pattern yang sama dari yang terkecil adalah 1 dan yang terbesar adalah agus. Dan menghabiskan waktu dari 0,11 waktu yang terkecil sampai dengan 0,75 ms (milisecond). Disini penulis akan mencoba mencari nilai tengah dari 10 sample diatas guna mencari tahu berapa vabyak waktu rata-rata yang dibutuhkan untuk mencari data peserta yang memiliki 400 peserta lebih. Dengan rumus menjumlahkan semua nilai *brute force* dan membaginya dari banyak sample yang dilakukan.

$$0,51 + 0,19 + 0,54 + 0,26 + 0,11 + 0,46 \\ + 0,24 + 0,40 + 0,84 + 0,75 \\ = 0,43 \text{ ms}$$

Dari hasil perhitungan sampel diatas didapatkan nilai 0,43 milisecond. Yang artinya rata-rata yang diperlukan untuk mencari sebuah peserta menggunakan algoritma *brute force* tidak mencapai 1 detik lebih tepatnya 0,43 ms. Tentunya sangat mudah penulis untuk mencari data tersebut.

4. Kesimpulan

Hasil dari penelitian ini adalah pembuatan *website* Pelatihan Internal ini sudah sesuai dengan penulis harapkan penggunaan Metode *Agile* sudah sangat tepat, Dan Metode Extreme Programming sebagai produk turunannya. Karena metode ini dalam pembuatannya sangat cepat dari fase *planning*, *design*, *coding* sampai dengan tahapan fase terakhir yaitu fase pengujian dan diulang Kembali ke fase *planning* hingga ketemu lagi ke fase pengujian terus menerus berulang secara *iterative*. Algoritma *Brute force* sangat membantu Admin dalam pencarian Data admin, Peserta, pelatihan, *self-learning*, *training calendar* dan lain-lain. Karena Algoritma ini sangat sederhana hanya mencari kata yang sesuai dengan admin inginkan. Dan Menampilkan semua data yang admin inginkan. Dan Rata – Rata waktu yang dibutuhkan dalam menggunakan algoritma *brute force* ini menghabiskan waktu 0,43 ms dalam setiap pattern yang dilakukan.

5. Daftar Pustaka

- [1] M. Ridwan, I. Fitri, and B. Benrahman, 2021. Rancang Bangun Marketplace Berbasis Website menggunakan Metodologi Systems Development Life Cycle (SDLC) dengan Model Waterfall. *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, 5, p.2. DOI: 10.35870/jtik.v5i2.209.
- [2] Khorasani, S.T. and Almasifard, M., 2017. Evolution of management theory within 20 century: A systemic overview of paradigm shifts in management. *International Review of Management and Marketing*, 7(3), pp.134-137.
- [3] Susan, E., 2019. Manajemen sumber daya manusia. *Adaara: Jurnal Manajemen Pendidikan Islam*, 9(2), pp.952-962.
- [4] Suri, M.I. and Puspaningrum, A.S., 2020. Sistem Informasi Manajemen Berita Berbasis Web. *Jurnal Teknologi Dan Sistem Informasi*, 1(1), pp.8-14.
- [5] Surya, A., Meliana Christianti, J. and Kom, S., 2021. Rancangan Aplikasi untuk Dokumen Penyelesaian Tugas Akhir. *Jurnal STRATEGI-Jurnal Maranatha*, 3(2), pp.345-352.
- [6] Chawla, T., 2017. An Agile Based Web Development Model for Cost Effective Website Design. 9(40), pp. 1125–1130.
- [7] Rasnaxis, A. and Berzisa, S., 2017. Method for adaptation and implementation of agile project management methodology. *Procedia Computer Science*, 104, pp.43-50. DOI: 10.1016/j.procs.2017.01.055.
- [8] Sudarsono, B.G., Lestari, S.P., Bani, A.U., Chandra, J. and Andry, J.F., 2020. Using an extreme programming method for hotel reservation system development. *Int. J. Emerg. Trends Eng. Res*, 8(6). DOI: 10.30534/ijeter/2020/01862020.
- [9] Silalahi, M. and Saragih, S.P., 2019. Implementasi Extreme Programming Dalam Perancangan Management Information System pada LP2M. *CESS (Journal of Computer Engineering, System and Science)*, 4(2), pp.208-216.
- [10] Khairul, K., Haryati, S. and Yusman, Y., 2018. Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. *Jurnal Teknologi Informasi dan Pendidikan*, 11(1), pp.1-6. DOI: 10.24036/tip.v11i1.102.
- [11] Pratiwi, H., Arfyanti, I. and Kurniawan, D., 2016. Implementasi Algoritma Brute Force dalam Aplikasi Kamus Istilah Kesehatan. *Jurnal Ilmiah Teknologi Infomasi Terapan*, 2(2). DOI: 10.33197/jitter.vol2.iss2.2016.99.

- [12] Suprpto, D.D.A., Fauziah, F., Fitri, I. and Hayati, N., 2020. Pengembangan Aplikasi Sistem Informasi Smart Register Online Berbasis Android Menggunakan Algoritma BruteForce. *Edumatic: Jurnal Pendidikan Informatika*, 4(1), pp.47-56. DOI: 10.29408/edumatic.v4i1.2106.
- [13] Violina, S., 2021. Analysis of brute force and branch & bound algorithms to solve the traveling salesperson problem (TSP). *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(8), pp.1226-1229.