

Analisa Penjadwalan Order Distribusi Produk di PT. Indofood CBP Sukses Makmur Medan Menggunakan Algoritma Genetika

1) **Lisa Febriani**

Universitas Labuhan Batu, Rantau Prapat, Indonesia

E-Mail: lisafebriani19@gmail.com

2) **Iwan Purnama**

Universitas Labuhan Batu, Rantau Prapat, Indonesia

E-Mail: lisafebriani19@gmail.com

3) **Budianto Bangun**

Universitas Labuhan Batu, Rantau Prapat, Indonesia

E-Mail: lisafebriani19@gmail.com

ABSTRACT

The existence of computers has even replaced many human workers, both in business and industry. This cannot be faulted because it is proven that the computerized system is much more efficient and profitable for the company than the manual system (with human labor). Almost every company, especially those that are developing and advancing, are currently competing to computerize their company's systems, to improve work efficiency and effectiveness. At PT. Indofood CBP Sukses Makmur Medan, especially those that produce goods, always occurs in the process of producing, storing, and selling products. With the application design carried out in the making of this research is the storage of input and output of data from existing goods, namely by managing all data on goods in the warehouse in a computerized system. Then the arrangement of incoming goods and outgoing goods in the warehouse rack pallets is arranged as optimally as possible by using the Genetic Algorithm method. The scheduling at PT. Indofood CBP Sukses Makmur Medan, which is a schedule of orders given to sales that can be on that day. Next, distribution orders are scheduled on the day, hour, and sales working. In optimizing the preparation of the distribution order schedule using the genetic algorithm, four parameters are needed, including the number of goals, the number of sales, the order crossover and mutation probability. In designing a product distribution order scheduling system at PT. Indofood CBP Sukses Makmur Medan consists of a login form, order data, sales data, destination data, time data, genetic algorithm processing and logout.

Keywords: Genetics, Scheduling, Distribution

PENDAHULUAN

Di dalam perusahaan komputer merupakan kebutuhan yang umum, sangat penting suatu unit komputer dalam sebuah kinerja perusahaan. Banyak hal menjadi lebih efisien dengan menggunakan sistem komputerisasi. Salah satu bentuk perusahaan yang sangat terbantu dengan adanya komputer adalah perusahaan di bidang industri[1][2].

Keberadaan komputer bahkan telah menggantikan banyak tenaga kerja manusia, baik di bidang bisnis maupun industri. Hal ini tidak bisa disalahkan karena memang terbukti bahwa sistem komputerisasi jauh lebih efisien dan menguntungkan bagi perusahaan dibanding dengan sistem manual (dengan tenaga kerja manusia). Hampir setiap perusahaan, khususnya yang sedang berkembang dan maju saat ini berlomba-lomba untuk mengkomputerisasi sistem yang dimiliki

perusahaannya, untuk meningkatkan efisiensi dan efektivitas kerja[3][4].

Pada perusahaan industri khususnya yang memproduksi suatu barang, selalu terjadi proses produksi, penyimpanan, dan penjualan hasil produksi. Lebih jauh lagi, pada pembuatan Penelitian ini penulis akan lebih memfokuskan pada permasalahan aplikasi penjadwalan order di dalam gudang[5][6].

Adapun perancangan aplikasi yang dilakukan dalam pembuatan Penelitian ini adalah penyimpanan pemasukan dan pengeluaran data-data dari barang yang ada yaitu dengan mengatur seluruh data-data barang yang berada di gudang dalam sistem komputerisasi[7][8]. Kemudian pengaturan barang yang masuk dan barang yang keluar di dalam palet rak gudang ditata seoptimal mungkin dengan menggunakan metode Algoritma Genetika.

METODE

1. Algoritma Genetika

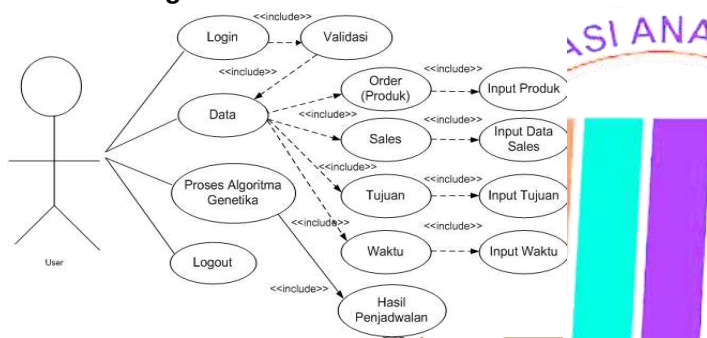
Algoritma Genetika[8][9] merupakan suatu metode heuristik yang dikembangkan berdasarkan prinsip genetika dan proses seleksi alamiah Teori Evolusi Darwin. Metode optimasi dikembangkan oleh John Holland sekitar tahun 1960-an dan dipopulerkan oleh salah seorang mahasiswanya, David Goldberg, pada tahun 1980-an. Proses pencarian penyelesaian atau proses terpilihnya sebuah penyelesaian dalam algoritma ini berlangsung sama seperti terpilihnya suatu individu untuk bertahan hidup dalam proses evolusi [9][10] Pada algoritma genetika pencarian dimulai dengan pembangkitan sejumlah "individu" secara acak yang disebut dengan kromosom. Kromosom-kromosom ini merupakan representasi calon penyelesaian yang akan diperiksa nilai yang sebenarnya.

2. Struktur Umum Algoritma Genetik

- a. Membangkitkan populasi awal
- b. Membentuk generasi baru
- c. Evaluasi solusi

HASIL DAN PEMBAHASAN

A. Perancangan



Gambar 1. Usecase perancangan sistem

B. Pembahasan

Berikut adalah contoh dari aplikasi algoritma genetika yang di gunakan untuk menyelesaikan masalah kombinasi. Misalkan persamaan $a+2b+3c+4d=20$, akan menentukan nilai a, b, c, dan d memenuhi persamaan diatas. Untuk menentukan pencarian pada metode algoritma genetika, untuk menyelesaikan masalah. Langkah-langkah dalam penyelesaian masalah dengan menggunakan algoritma genetika adalah sebagai berikut:

1. Pembentukan *Chromosome*

Sehubungan pencarian yang digunakan adalah a, b, c, dan d maka penentuan variabel a, b, c, dan d di bentuk menjadi gen-gen pembentuk *chromosome*. Jika gen a dimisalkan adalah sales, b dimisalkan adalah order distribusi, c dimisalkan adalah hari/waktu dan d dimisalkan adalah Tujuan. Untuk penentuan a adalah 0

sampai 20, b adalah 0 sampai 6, c adalah 0 sampai 6 dan d adalah 1 sampai 9.

2. Inisialisasi

Proses inisialisasi ini dilakukan dengan cara memberikan nilai awal gen-gen dengan nilai acak sesuai batasan yang telah di tentukan sebelumnya.

Misalkan ditentukan jumlah populasi adalah 6, maka:

$$Chromosome [I] = [a;b;c;d] = [5;2;3;4]$$

$$Chromosome [II] = [a;b;c;d] = [1;3;4;6]$$

$$Chromosome [III] = [a;b;c;d] = [2;1;2;8]$$

$$Chromosome [IV] = [a;b;c;d] = [3;3;1;7]$$

$$Chromosome [V] = [a;b;c;d] = [4;1;6;3]$$

$$Chromosome [VI] = [a;b;c;d] = [6;2;5;5]$$

3. Evaluasi *Chromosome*

Permasalahan yang ingin diselesaikan adalah nilai variabel a, b, c, dan d yang akan memenuhi permasalahan $a+2b+3c+4d=20$, maka fungsi objektif akan digunakan untuk mendapatkan solusi adalah fungsi objektif (*chromosome*) = $[(a+2b+3c+4d)-20]$ maka penentuan solusinya adalah sebagai berikut sesuai dengan penentuan chromosome yang diatas;

$$Fungsi_objektif(chromosome[I]) = Abs ((5 + 2*2 + 3*3 + 4*4) -20)$$

$$= Abs ((5+4+9+16) -20)$$

$$= Abs (34-20)$$

$$= 14$$

$$Fungsi_objektif(chromosome[II]) = Abs ((1 + 2*3 + 3*4 + 4*6) -20)$$

$$= Abs ((1+6+12+24) -20)$$

$$= Abs (43-20)$$

$$= 23$$

$$Fungsi_objektif(chromosome[III]) = Abs ((2 + 2*1 + 3*2 + 4*8) -20)$$

$$= Abs ((2+2+6+40) -20)$$

$$= Abs (56-20)$$

$$= 36$$

$$Fungsi_objektif(chromosome[IV]) = Abs ((3 + 2*3 + 3*1 + 4*7) -20)$$

$$= Abs ((3+6+3+28) -20)$$

$$= Abs (40-20)$$

$$= 20$$

$$Fungsi_objektif(chromosome[V]) = Abs ((4 + 2*1 + 3*6 + 4*3) -20)$$

$$= Abs ((4+2+18+12) -20)$$

$$= Abs (38-20)$$

$$= 18$$

$$Fungsi_objektif(chromosome[VI]) = Abs ((6 + 2*2 + 3*5 + 4*5) -20)$$

$$= Abs ((6+4+15+20) -20)$$

$$= Abs (45-20)$$

$$= 25$$

Rata-rata dari fungsi objektif adalah:

$$Rata-rata = (14+23+36+20+18+25) /6$$

$$= 136/6$$

$$=22.6$$

4. Seleksi *Chromosome*

Pada proses seleksi ini dilakukan dengan cara membuat *chromosome* yang mempunyai fungsi_objektif kecil yang kemungkinan terpilih besar atau mempunyai nilai probabilitas yang tinggi. Untuk itu di perlukan perhitungan $fitness = 1/(1+fungsi_objektif)$, fungsi objektif perlu ditambah 1 untuk menghindari kesalahan program yang diakibatkan pembagian oleh 0.

$$Fitness [1] = 1/(fungsi_objektif[1]+1) \\ = 1/15 \\ = 0.066$$

$$Fitness [2] = 1/(fungsi_objektif [2]+1) \\ = 1/24 \\ = 0.041$$

$$Fitness [3] = 1/(fungsi_objektif[3]+1) \\ = 1/37 \\ = 0.027$$

$$Fitness [4] = 1/(fungsi_objektif[4]+1) \\ = 1/21 \\ = 0.047$$

$$Fitness [5] = 1/(fungsi_objektif[5]+1) \\ = 1/19 \\ = 0.052$$

$$Fitness [6] = 1/(fungsi_objektif[6]+1) \\ = 1/26 \\ = 0.038$$

$$Total fitness = \\ 0.066+0.041+0.027+0.047+ 0.052+ 0.038 \\ = 0.271$$

Rumus untuk mencari probabilitas : $P[i] = fitness[i] / total_fitness$

$$P[1] = 0.066 / 0.271 \\ = 0.243$$

$$P[2] = 0.041 / 0.271 \\ = 0.151$$

$$P[3] = 0.027 / 0.271 \\ = 0.099$$

$$P[4] = 0.047 / 0.271 \\ = 0.173$$

$$P[5] = 0.052 / 0.271 \\ = 0.191$$

$$P[6] = 0.038 / 0.271 \\ = 0.140$$

Dari probabilitas diatas maka dapat ditemukan nilai tertinggi dari pencarian diatas adalah pada probabilitas 1 yang lebih besar. Maka dari penentuan *chromosome* tersebut mempunyai probabilitas untuk terpilih pada generasi selanjutnya lebih besar dari *chromosome* lainnya. Untuk proses seleksi di gunakan *roulette wheel*, supaya untuk menentukan nilai kumulatif probabilitasnya:

$$C[1] = 0.243$$

$$C[2] = 0.243 + 0.151 = 0.394$$

$$C[3] = 0.243 + 0.151 + 0.099 = 0.493$$

$$C[4] = 0.243 + 0.151 + 0.099 + 0.173 = 0.666$$

$$C[5] = 0.243 + 0.151 + 0.099 + 0.173 + 0.191 = 0.857$$

$$C[6] = 0.243 + 0.151 + 0.099 + 0.173 + 0.191 + 0.140 = 1$$

Setelah dilakukan perhitungan komulative probabilitasnya maka proses seleksi menggunakan *roulette wheel* dapat dilakukan. Prosesnya adalah dengan membangkitkan bilangan acak R dalam range 0-1. Jika $R[k] < C[1]$ maka pilih *chromosome* ke-k sebagai induk dengan syarat $V[k-1] < R < C[k]$. Jika diputar *roulette wheel* sebanyak jumlah populasi yaitu 6 kali (bangkitkan bilangan acak R) dan pada tiap putaran, maka dipilih satu *chromosome* untuk populasi baru, misalnya:

$$R[1] = 0.260$$

$$R[2] = 0.201$$

$$R[3] = 0.100$$

$$R[4] = 0.180$$

$$R[5] = 0.225$$

$$R[6] = 0.150$$

Angka acak pertama R[1] adalah lebih besar dari C[1] dan lebih kecil dari pada C[2] maka pilih *chromosome*[1] sebagai *chromosome* baru hasil proses seleksi adalah :

$$Chromosome[1] = chromosome[1]$$

$$Chromosome[2] = chromosome[1]$$

$$Chromosome[3] = chromosome[5]$$

$$Chromosome[4] = chromosome[1]$$

$$Chromosome[5] = chromosome[3]$$

$$Chromosome[6] = chromosome[2]$$

Chromosome baru hasil proses seleksi adalah:

$$Chromosome[1] = [5;2;3;4]$$

$$Chromosome[2] = [5;2;3;4]$$

$$Chromosome[3] = [4;1;6;3]$$

$$Chromosome[4] = [4;1;6;3]$$

$$Chromosome[5] = [2;1;2;8]$$

$$Chromosome[6] = [1;3;4;6]$$

5. Crossover

Setelah proses seleksi maka proses selanjutnya adalah proses *crossover*. Metode ini menggunakan salah satu *one-cut point*, yaitu memilih secara acak satu posisi dalam *chromosome* induk kemudian saling menukar gen. *Chromosome* yang dijadikan induk dipilih secara acak dan jumlah *chromosome* yang mengalami *crossover* dipengaruhi oleh parameter *crossover_rate* (pc).

Setelah melakukan pemilihan induk proses selanjutnya adalah menentukan posisi *crossover*. Ini dilakukan dengan cara dengan batasan 1 sampai (panjang *chromosome*-1), dalam kasus ini bilangan acak yang dibangkitkan adalah 1-3. Misalkan didapatkan potongan gen tersebut saling ditukarkan antar induk.

$$Chromosome [1] > < Chromosome [3]$$

Chromosome [3] >< Chromosome [5]
 Chromosome [5] >< Chromosome [1]
 Offspring[1] = chromosome [1] >< chromosome [3]

$$= [5;2;3;4] >< [2;1;2;8]$$

$$= [5;1;3;8]$$

Offspring[3] = chromosome [3] >< chromosome [5]

$$= [2;1;2;8] >< [4;1;6;3]$$

$$= [2;1;2;3]$$

Offspring[5] = chromosome [5] >< chromosome [1]

$$= [5;2;3;4] >< [5;2;3;4]$$

$$= [5;2;3;4]$$

Dengan demikian populasi chromosome setelah mengalami proses crossover menjadi :

Chromosome [1] = [5;1;3;2]

Chromosome [2] = [1;3;4;6]

Chromosome [3] = [5;1;3;2]

Chromosome [4] = [3;3;1;7]

Chromosome [5] = [5;2;3;4]

Chromosome [6] = [6;2;5;5]

6. Mutasi

Jumlah chromosome yang mengalami mutasi dalam satu populasi ditentukan oleh parameter *mutation_rate*. Proses mutasi dilakukan dengan cara mengganti satu gen yang terpilih secara acak dengan suatu nilai baru yang didapatkan secara acak. Prosesnya adalah sebagai berikut :

Pertama hitung terlebih dahulu panjang total gen yang ada dalam satu populasi.

Dalam kasus ini panjang total gen adalah $total_gen = (\text{jumlah gen dalam chromosome}) \times \text{jumlah populasi}$

$$= 4 \times 6$$

$$= 24$$

Dan untuk menentukan posisi gen yang mengalami mutasi dengan cara membangkitkan bilangan integer acak antara 1 sampai $total_gen$, yaitu 1 sampai 26. Jika bilangan acak yang dibangkitkan lebih kecil daripada variabel *mutation_rate* (pm) maka pilih posisi tersebut sebagai sub-chromosome yang mengalami mutasi. Misal pm ditentukan 10% maka diharapkan ada 10% dari $total_gen$ yang mengalami populasi :

$$\text{Jumlah mutasi} = 0.1 \times 24$$

$$= 2.4$$

$$= 2$$

Proses yang sama ini akan di lanjutkan pada generasi sebelumnya yaitu proses evaluasi, seleksi, crossover dan mutasi yang kemudian akan menghasilkan chromosome-chromosome baru untuk generasi selanjutnya. Setelah dilakukan perhitungan yang diatas maka di pilihlah chromosome yang terbaik yaitu:

Chromosome = [05;01;03;02]

Jika dikodekan maka:

$a=05$; $b=01$; $c=03$; $d=02$

jika dihitung terhadap persamaan $f=a+2b+3c+4d$:

$$5 + (2 \times 1) + (3 \times 3) + (4 \times 2) = 24$$

Dari pencarian di atas dapat diketahui hasil akhir dari perhitungan yaitu;

Chromosome [1] = [5;1;3;2]

Chromosome [2] = [1;3;4;6]

Chromosome [3] = [5;1;3;2]

Chromosome [4] = [3;3;1;7]

Chromosome [5] = [5;2;3;4]

Chromosome [6] = [6;2;5;5]

Chromosome [1] = [Indra; Indomie; Senin; Sanggal]

Chromosome [2] = [Kartini; Kecap Indofood; Selasa; Petisah]

Chromosome [3] = [Indra; Indomie; Senin; Medan Baru]

Chromosome [4] = [Herman; Kecap Indofood; Rabu; Helvetia]

Chromosome [5] = [Indra; Chitato; Senin; Petisah]

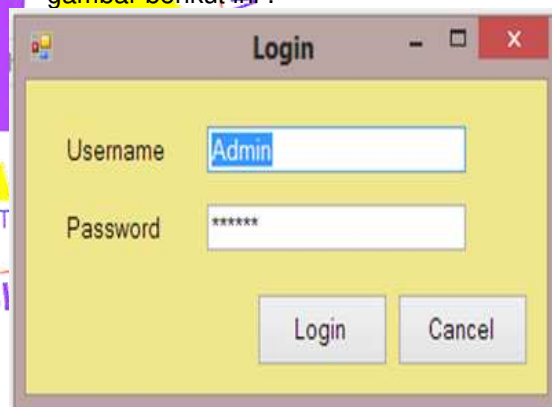
Chromosome [6] = [Jonny Manalu; Chitato; Kamis; Polonia]

C. Implementasi Sistem

Pada penelitian ini, sistem dibangun menggunakan *Visual Basic.Net 2008*, berikut merupakan implementasi sistem dari langkah pertama hingga penentuan solusi.

1. Tampilan Form Login

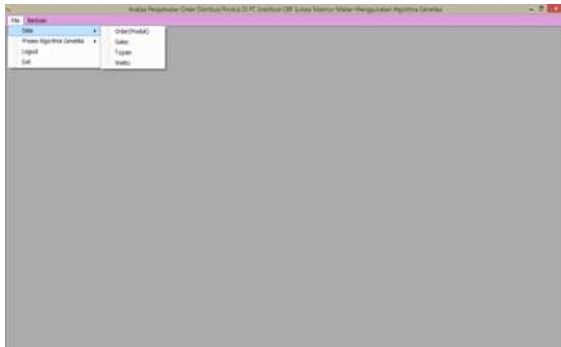
Tampilan ini merupakan tampilan dari form login dimana berfungsi untuk melakukan proses login. Yang dapat dilihat pada gambar berikut ini :



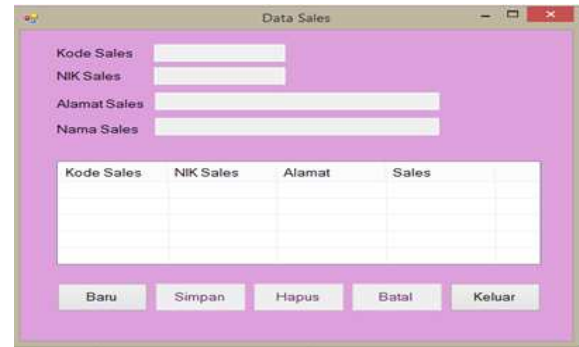
Gambar 1. Tampilan Form Login

2. Tampilan Form Menu Utama

Berikut ini merupakan tampilan dari form menu utama. Yang berfungsi untuk pemanggilan form-form lainnya pada sistem.



Gambar 2. Tampilan *Form* Menu Utama



Gambar 5. Tampilan Form Data Sales

3. Tampilan Form Order distribusi

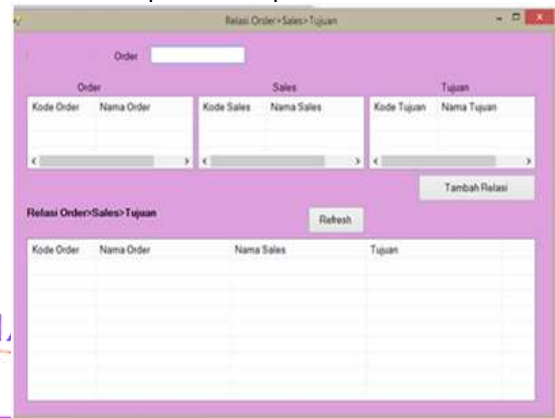
Form order distribusi merupakan form yang berfungsi untuk menginputkan data order distribusi. Berikut merupakan tampilan dari form order distribusi.



Gambar 3. Tampilan Form Order distribusi

6. Tampilan Form Relasi

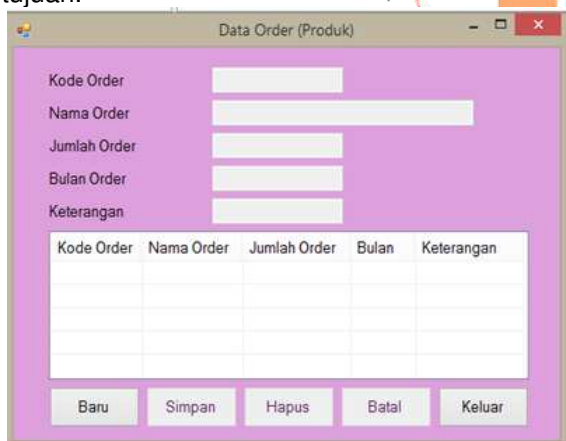
Form Relasi berfungsi untuk menampilkan relasi antara order distribusi, sales, tujuan. Berikut merupakan tampilan dari form relasi.



Gambar 6. Tampilan Form Perhitungan Vector

4. Tampilan Form Data Tujuan

Form alternatif merupakan form yang berfungsi untuk penginputan data tujuan yang ada. Berikut merupakan tampilan dari form data tujuan.



Gambar 4. Tampilan Form Data Tujuan

7. Tampilan Form Penjadwalan Order Distribusi Produk Dengan Algoritma Genetika.

Form ini berfungsi untuk melakukan proses penjadwalan berdasarkan algoritma genetika. Berikut ini merupakan tampilan dari form penjadwalan Order Distribusi Produk dengan algoritma genetika.



Gambar 7. Tampilan Form Penjadwalan

5. Tampilan *Form* Data Sales

Form Data Sales berfungsi untuk penginputan data sales. Berikut merupakan tampilan dari form data sales.

8. Tampilan Form hasil Penjadwalan

Form hasil penjadwalan berfungsi untuk menampilkan hasil penjadwalan berdasarkan algoritma genetika. Berikut ini merupakan tampilan form hasil pejadwalan.

ID Gen	No. Gen	Kromosom	Fitness
1	[a,b,c,d]	I	[3,1,2,7]
2	[a,b,c,d]	II	[2,2,4,5]
3	[a,b,c,d]	III	[5,3,7,9]
4	[a,b,c,d]	IV	[6,1,5,4]
5	[a,b,c,d]	V	[4,3,6,2]
6	[a,b,c,d]	VI	[1,2,3,6]

Gambar 8. Tampilan Form Hasil Penjadwalan

KESIMPULAN

Dari hasil penelitian yang dilakukan dapat disimpulkan bahwa penjadwalan pada PT. Indofood CBP Sukses Makmur Medan yaitu jadwal order diberi kepada sales yang bisa pada hari tersebut. Selanjutnya, order distribusi di jadwalkan pada hari, jam, dan sales yang bekerja tersebut. Dalam mengoptimalkan penyusunan jadwal order distribusi menggunakan algoritma genetika diperlukan empat parameter antara lain jumlah tujuan, jumlah sales, order probabilitas crossover dan mutasi. Dalam perancangan sistem penjadwalan order distribusi produk di PT. Indofood CBP Sukses Makmur Medan terdiri dari form login, data order, data sales, data tujuan, data waktu, proses algoritma genetika dan logout.

DAFTAR PUSTAKA

- [1]. M. N. K. Nababan, "PENERAPAN WEB BASED APPLICATION SISTEM PENILAIAN (REPORTING) SISWA," vol. 3, no. 2, pp. 60–64, 2020.
- [2]. J. Wijaya, V. Frans, and F. Azmi, "Aplikasi Traveling Salesman Problem Dengan GPS dan Metode Backtracking," vol. 3, no. 2, pp. 81–90, 2020.
- [3]. D. Sitanggang, S. Simangunsong, R. U. Sipayung, and A. S. Nababan, "Perancangan Aplikasi Penyeleksian Penerimaan Siswa Untuk Mengikuti Olimpiade Sains Berbasis Android," vol. 3, no. 2, pp. 34–43, 2020.
- [4]. M. N. K. Nababan, T. Desyana, S. Rumapea, S. S. Siotang, and L. M. Gultom, "ANDROID MENGGUNAKAN ALGORITMA AES," vol. 3, no. 2, pp. 76–80, 2020.
- [5]. M. A. Iqbal and R. Rosnelly, "Perancangan Aplikasi Media Pembelajaran Pengenalan Lapisan Bumi Menggunakan Augmented Reality Berbasis Android," *J. Mhs. Fak.*, vol. 3, no. 2, pp. 26–33, 2020, [Online]. Available: <http://e-journal.potensi-utama.ac.id/ojs/index.php/FTIK/article/view/>

- [6]. [D. A. Butar-butur, D. Amalia, K. Mayra, A. [1] T. Limbong, J. Simarmata, A. Fauzi, P. Siagian, and P. Tarigan, "Optimization of Employee Assignment in Content Management System Making With Hungarian Method," 2018.
- [7]. Nst, and Y. Naibaho, "Pemanfaatan Teknologi Informasi Dalam Pengambilan Keputusan Penilaian Karyawan Terbaik," vol. 2, no. 1, pp. 43–46, 2020.
- [8]. J. Banjarnahor and A. X. Lim, "Aplikasi Pembayaran Uang Kuliah Pada Universitas Prima Indonesia Menggunakan Metode Fuzzy Logic Berbasis Android," vol. 2, no. 1, pp. 7–13, 2020.
- [9]. D. Sitanggang *et al.*, "ANALISIS PERANCANGAN APLIKASI PEMESANAN TIKET KAPAL BERBASIS ANDROID," vol. 2, no. 1, pp. 34–38, 2020.
- [10]. O. Sihombing, S. Sihombing, M. L. Pasaribu, R. Kris, and D. Saragih, "Website Rekomendasi Tempat Kuliner dengan Metode Social Trust Path," vol. 3, no. 1, pp. 1–6, 2020.
- [11]. S. P. Tamba, D. R. Hia, D. Prayitna, and A. Tryvaldy, "Pemanfaatan Teknologi Berbasis Mobile Untuk Manajemen Kontrol Nilai Dan Absensi Siswa Pada Mts Al-Ittihadiyah Medan," vol. 2, no. 1, pp. 18–22, 2020.
- [12]. F. A. Nugroho, "Perancangan Sistem Pakar Diagnosa Penyakit Jantung dengan Metode Forward Chaining," *J. Inform. Univ. Pamulang*, vol. 3, no. 2, p. 75, 2018, doi: 10.32493/informatika.v3i2.1431.