Pemodelan *Gantt diagram* dengan Struktur Data *Binding* untuk Masalah Penjadwalan Proyek Perangkat Lunak

Noviar Iriansya*[1], Sugiarto Cokrowibowo[2], Arnita Irianti[3], Wawan Firgiawan[4]
[1,2,3]Program Studi Teknik Informatika, Universitas Sulawesi Barat
E-mail: [1]noviaririansya2098@gmail.com, [2]sugiarto.cokrowibowo@unsulbar.ac.id,
[3]arnitairianti@unsulbar.ac.id, [4]wawanfirgiawan@unsulbar.ac.id

Abstrak

Manajemen proyek adalah sebuah aktivitas kunci perusahaan atau suatu organisasi dalam pengembangan sebuah proyek. Kesulitan utama dalam aktivitas ini adalah sulitnya membagi alokasi pekerja terhadap pekerjaan yang ada sehingga akan berakibat pada durasi dan biaya pengerjaan proyek yang dikeluarkan nantinya. Masalah inilah yang disebut dengan Software Project Scheduling Problem (SPSP) atau masalah penjadwalan proyek perangkat lunak. Tujuan dari penelitan ini adalah untuk memodelkan solusi ke dalam bentuk Gantt diagram untuk memudahkan pembacaan urutan task/pekerjaan yang harus. Dengan struktur data binding, gantt diagram dapat dibangun mengikuti aturan dari task precedence graph sehingga urutan dan waktu pengerjaan task dapat menyesuaikan secara otomatis. Dari gantt diagram ini kemudian penjadwal dapat melihat data urutan task/tugas yang ada sehingga pengerjaan tugas dapat diselesaikan dengan tepat dan efisien.

Kata kunci : *Software Project Scheduling Problem, SPSP, Gantt diagram*

Abstract

Project management is a key activity of a company or organization in developing a project. The main burden in this activity is the difficulty of dividing the allocation of workers to existing work, which will have an impact on the duration and costs of project work that will be incurred later. This problem is called the Software Project Scheduling Problem (SPSP) or software project scheduling problem. The aim of this research is to model the solution in the form of a Gantt diagram to make it easier to read the sequence of tasks/work that must be done. With a binding data structure, a Gantt chart can be built following the rules of the task priority chart so that the sequence and time for completing tasks can be adjusted automatically. From this Gantt diagram, the scheduler can see the existing task sequence data so that task work can be completed correctly and efficiently.

Keywords : *Software Project Scheduling Problem, SPSP, Gantt diagram*

1. PENDAHULUAN

Manajemen proyek adalah sebuah aktivitas kunci dalam perusahaan atau sebuah organisasi pengembangan proyek proyek [1] sebab jika aktivitas ini tidak dijalankan dengan baik, maka akan berpengaruh pada waktu pengerjaan dan biaya yang terkait dengan proyek [2].

Salah satu tujuan utama dalam perencanaan suatu proyek adalah meminimalkan biaya dan durasi proyek yang biasanya saling bertentangan, karena ketika satu dikurangi, yang lain biasanya meningkat [3]. Salah satu dari jenis proyek ini adalah *Software project scheduling problem (SPSP)* atau Masalah Penjadwalan Proyek Perangkat Lunak.

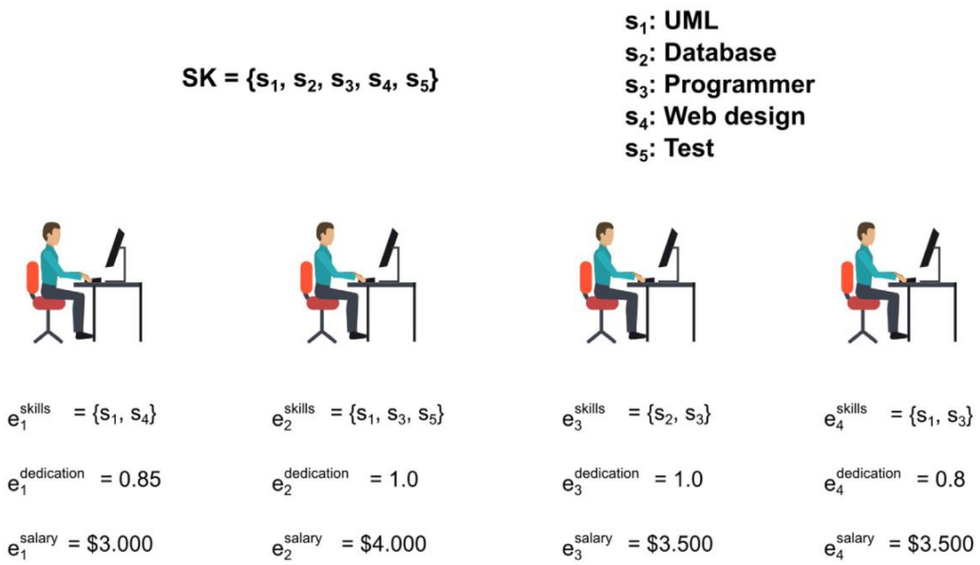
SPSP atau Software project scheduling problem adalah masalah Penjadwalan Proyek perangkat lunak yang terdiri dari penjadwalan dan penugasan para pekerja dengan tugas-tugas yang ada dalam waktu terbatatas sehingga dapat meminimalkan biaya dan durasi untuk keseluruhan proyek [4]. Oleh karena itu, penelitian ini mencoba menerapkan suatu bentuk pemodelan solusi dalam pemodelan *Gantt diagram* menggunakan struktur data *binding* untuk menyelesaikan masalah penjadwalan proyek perangkat lunak. Visualisasi *gantt* diagram digunakan dengan tujuan untuk memudahkan pembacaan data pemodelan solusi untuk keseluruhan proyek pada SPSP ini.

Berdasarkan latar belakang yang diuraikan, maka terdapat hal yang menjadi permasalahan penelitian ini, yaitu: Bagaimana pemodelan *Gantt diagram* dengan struktur data *binding* dalam menyelesaikan masalah penjadwalan proyek perangkat lunak (SPSP).

2. METODE

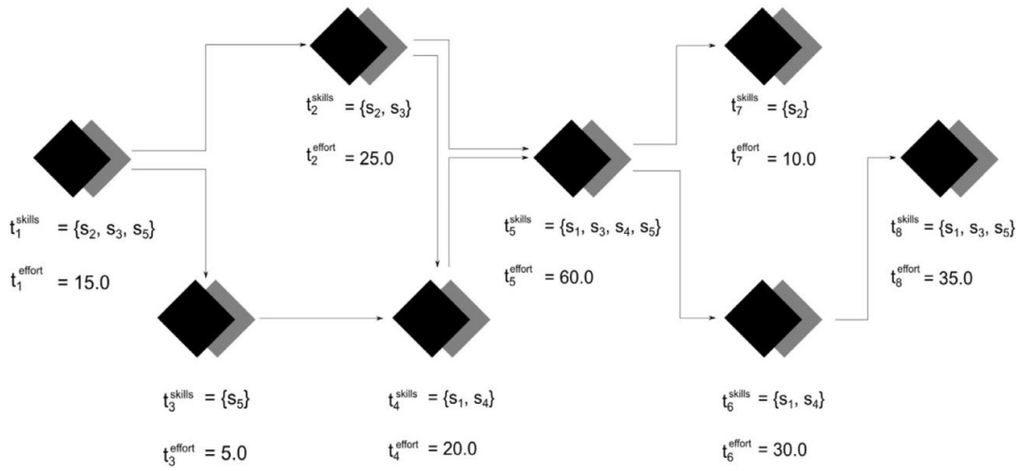
2.1. Software Project Scheduling Problem

SPSP atau *Software project scheduling problem* adalah masalah Penjadwalan Proyek perangkat lunak yang secara spesifik terdiri dari penjadwalan dan penugasan para pekerja dengan tugas-tugas yang ada sehingga dapat meminimalkan biaya dan durasi untuk keseluruhan proyek, dan juga prioritas tugas dan batasan sumber daya terpenuhi [3].



Gambar 1 Contoh tim pekerja dengan *skill* masing-masing [5]

SPSP juga mempertimbangkan tujuan tambahan untuk meminimalkan biaya proyek, yang diukur dengan biaya setiap karyawan yang dialokasikan untuk proyek tersebut. Setiap karyawan memiliki beberapa kemungkinan keterampilan dan dia dapat melakukan banyak tugas dalam satu hari kerja [6].



Gambar 2 Contoh sebuah *task precedence graph* [6]

Hubungan antar tugas dimodelkan sebagai *Task Precedence Graph* (TPG). Ini adalah graf berarah dengan himpunan *vertex T = {t1, t2..., t|T|}* yang direpresentasikan sebagai tugas. Sebuah tanda panah (t_i, t_j) ∈ A berarti bahwa tugas t_i harus diselesaikan sebelum tugas t_j dapat dimulai [6].

Solusi untuk masalah tersebut direpresentasikan oleh matriks X = (Xij) dengan ukuran |E| × |T| di mana Xij ≥ 0, dan E adalah himpunan karyawan. Setiap elemen Xij adalah derajat dedikasi karyawan ei terhadap tugas t_j. Jadi, jika karyawan e_i melaksanakan tugas t_j dengan tingkat dedikasi 0.8, ini berarti dia menghabiskan 80% dari hari kerjanya untuk tugas tersebut. Jika derajat pengabdian adalah 0, maka karyawan tersebut tidak mengerjakan tugas tersebut.

Tabel 1 Contoh *dedication degree matriks* (matriks Xij) [5].

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
e_1	0.00	0.00	0.00	0.80	0.40	0.20	0.00	0.50
e_2	1.00	0.25	0.75	0.25	0.80	0.30	0.00	0.35
e_3	0.50	0.60	0.00	0.00	1.00	0.00	1.00	0.40
e_4	0.75	0.45	0.00	0.60	0.30	0.75	0.00	0.50

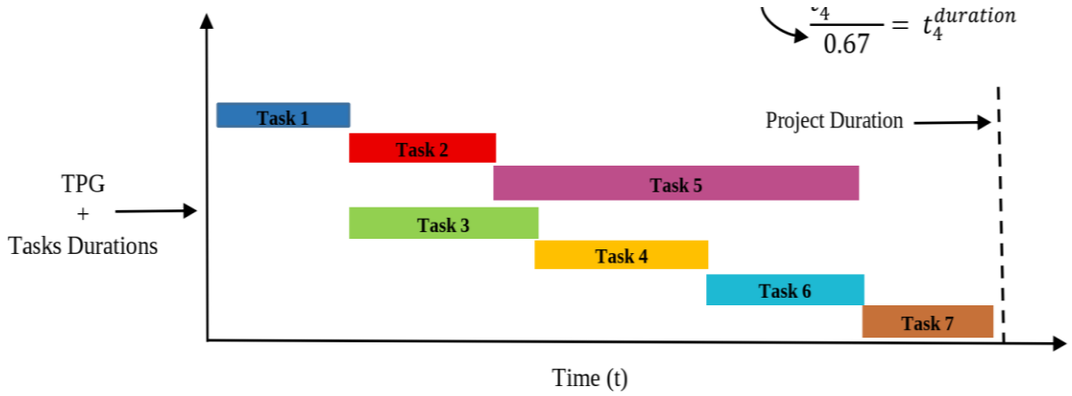
Informasi Matriks Xij disajikan pada Tabel di atas dan digunakan untuk menghitung durasi setiap tugas, serta waktu mulai dan selesai tugas [6]. Untuk menghitung durasi proyek, terlebih dahulu durasi setiap tugas dihitung dengan:

$$t_j^{dur} = \frac{t_j^{effort}}{\sum_{i=1}^{|E|} x_{ij}} \quad (2.1)$$

Biaya proyek adalah jumlah biaya yang dibayarkan kepada karyawan atas dedikasi mereka terhadap proyek, yang diberikan dengan mengalikan gaji karyawan dengan waktu yang dihabiskan untuk proyek tersebut. Kali ini adalah jumlah dedikasi dikalikan dengan durasi setiap tugas [6].

$$p_{cost} = \sum_{i=1}^{|E|} \sum_{j=1}^{|T|} e_i^{salary} \cdot x_{ij} \cdot t_j^{dur} \quad (2.2)$$

Setelah itu, berdasarkan dari TPG, maka akan didapatkan hasil akhir dengan sebuah model *Gantt diagram.*



Gambar 3 Contoh model *Gantt diagram*[7].

2.2 Struktur Data *Binding*

Data *Binding* atau pengikatan data adalah proses yang menggabungkan dua sumber data dan menyinkronkannya. Dengan pengikatan data, perubahan pada elemen dalam kumpulan data secara otomatis diperbarui dalam kumpulan data terikat [8].

Setiap perubahan data dalam satu kumpulan data secara otomatis tercermin dalam kumpulan data terikat lainnya. Dalam sintaksis yang mengikat, sumber data adalah penyedia data, dan kumpulan data lainnya adalah konsumen data. Pengikatan tersebut membentuk hubungan antara penyedia data dan konsumen data, memungkinkan koneksi antara data elemen visual dan sumber data.

2.3 Task Precedence Graph

Task Precedence Graph merupakan grafik yang mengatur suatu pekerjaan ntuk menentukan urutan tugas yang harus diselesaikan, penjadwalan tugas juga menggunakan grafik prioritas. Grafik ini memungkinkan penjadwal untuk mengoptimalkan pelaksanaan tugas dan meningkatkan kinerja sistem dengan mengungkapkan pekerjaan mana yang dapat diselesaikan secara bersamaan dan mana yang harus menunggu pekerjaan lain selesai [9]. *Task precedence graph* ini merupakan faktor yang menentukan bagaimana hasil visualisasi *gantt* diagram nantinya, visualisasi yang dihasilkan nanti akan menentukan pekerjaan mana saja yang harus diselesaikan terlebih dahulu, maupun pekerjaan mana saja yang bisa dikerjakan secara bersamaan, sebab hal ini akan berdampak pada durasi setiap pekerjaan, serta durasi dan biaya keseluruhan proyek.

2.4 *Gantt diagram*

Gantt diagram adalah diagram batang horizontal yang digunakan untuk menganalisis efisiensi proses dan digunakan secara historis di sektor manufaktur untuk melacak tugas-tugas tertentu dalam jadwal proyek dan untuk mencapai visibilitas yang lebih baik mengenai kendala alur kerja untuk analisis perbaikan proses, misalnya kemacetan dalam laju pemrosesan atau waktu tunggu [10].

Pada penelitian ini, solusi yang dihasilkan dalam menyelesaikan masalah penjadwalan proyek perangkat lunak digambarkan dalam visualisasi *Gantt diagram* sebagaimana pada gambar 2.3 di atas. Visualisasi ke dalam model gambar bertujuan untuk memudahkan pembacaan urutan *task* (pekerjaan), *task duration* (durasi pekerjaan), *project duration* (durasi proyek) dan *project cost* (biaya proyek).

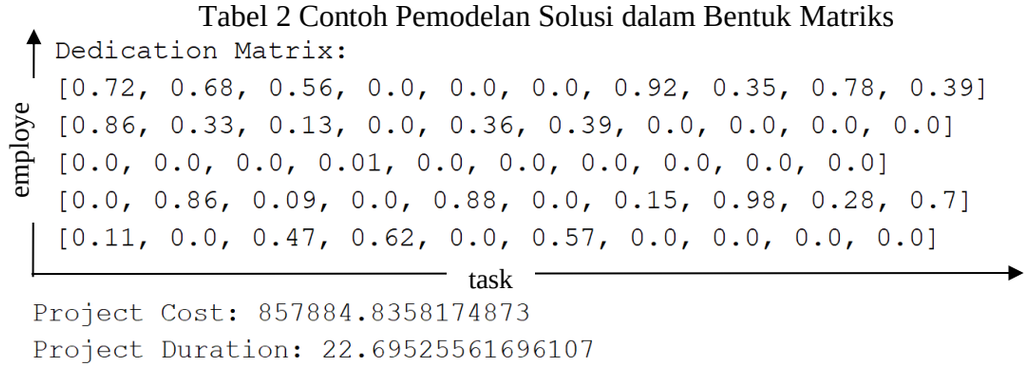
3. HASIL DAN PEMBAHASAN

3.1 Model Representasi Solusi

3.1.1 Model Solusi 1

Pada penelitian ini, dalam memodelkan hubungan antara employee dan *task* kemudian digambarkan dalam bentuk matriks dengan ukuran M×N dimana M adalah banyak *employee* dan N adalah banyaknya *task*, dengan elemen matriks yang diisi dengan nilai random dari rentang 0-1 sebagai representasi dari nilai dedikasi kerja setiap *employee* terhadap *task*. Pemodelan dilakukan dalam bentuk *array* 2 dimensi. *Array* tersebut memuat *task-task* tiap proses. Dapat dilihat pada tabel matriks berikut:

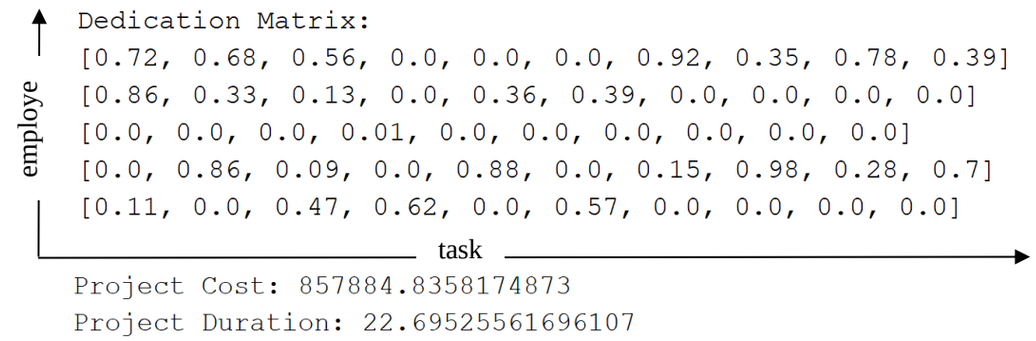
Tabel 2 Contoh Pemodelan Solusi dalam Bentuk Matriks



pada tabel 1 digambarkan sebuah contoh solusi dengan membangkitkan nilai *dedication* (nilai elemen matriks) secara acak yang dimodelkan dalam bentuk matriks/*array* 2 dimensi dimana jumlah baris sama dengan jumlah *employee* dan jumlah kolom sama dengan jumlah *task*. Dari matriks Xij yang sudah dibangkitkan tersebut kemudian menghasilkan *project cost* (biaya proyek), *project duration* (durasi proyek).

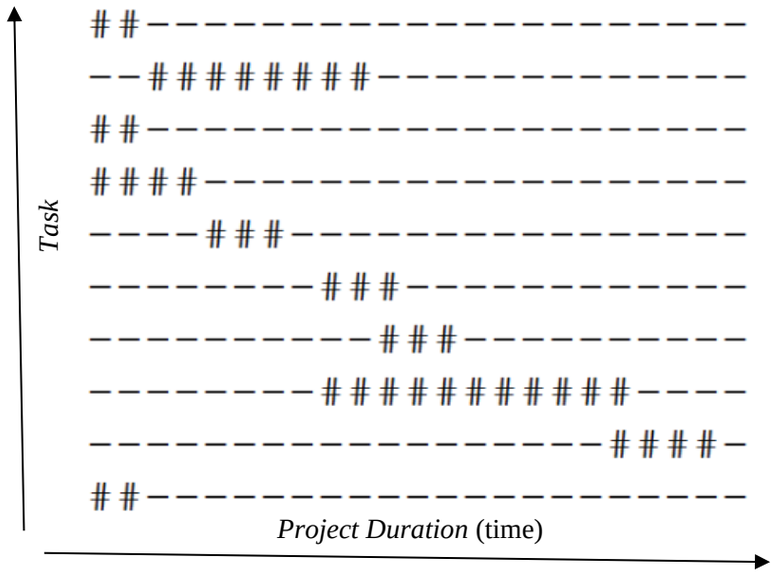
3.1.2 Model Solusi 2

Model matriks yang telah dibangkitkan kemudian diubah ke dalam bentuk *Gantt diagram* dengan mengikuti aturan *task precedence graph* dari data proyek yang ada dengan menggunakan teknik struktur data *binding*.



Tabel 3 Contoh Pemodelan Solusi dalam Bentuk Matriks

Gambar 4 di bawah merupakan *Gantt diagram* hasil viualisasi dari model matriks yang sebelumnya sudah dibangkitkan dengan mengikuti aturan *task precedence graph* menggunakan struktur data *binding*.



Gambar 4 hasil visualisasi *Gantt diagram* dari matriks Xij

Sesuai dengan gambar 4 di atas, setiap baris *task* memiliki *task duration* yang merupakan waktu yang dibutuhkan untuk mengerjakan keseluruhan proyek. Dan *Gantt diagram* merupakan visualisasi dari skenario pengerjaan proyek, dengan jumlah baris sebanyak jumlah *task* sedangkan kolom yang digambarkan dengan tanda (#) sebagai panjang *task duration* atau durasi setiap *task* dan (-) sebagai panjang dari *project duration* atau keseluruhan proyek.

3.2 *Pseudo-Code* Pemodelan *Gantt diagram*

Visualisasi *gantt* diagram dalam pemodelan solusi masalah penjadwalan proyek perangkat lunak dibuat secara otomatis menggunakan program. Program tersebut dibuat dari pseudo-code berikut:

```
Class TaskVertex
  Properties:
      String name
      double duration
      double startTime
      double endTime
      ArrayList<TaskVertex> leftTaskVertex = null
      ArrayList<TaskVertex> rightTaskVertex = null

  Method addLeft(TaskVertex taskVertex):
    If leftTaskVertex is null:
      Create a new ArrayList and assign it to leftTaskVertex
    Add taskVertex to leftTaskVertex

  Method addRight(TaskVertex taskVertex):
    If rightTaskVertex is null:
      Create a new ArrayList and assign it to rightTaskVertex
    Add taskVertex to rightTaskVertex

  Method getStartTime():
    Return startTime
```

```
Method getEndTime():
   Return endTime

Method getDuration():
   Return duration

Method updateStartTime(double newStartTime):
   Set startTime to newStartTime
   Set endTime to startTime + duration
   If right Task Vertex is not null:
      For each task Vertex in right Task Vertex:
         maxEndTime = task Vertex.getMaxEndTimeFromLeft Task Vertex()
         task Vertex.updateStartTime(maxEndTime)

Method getMaxEndTimeFromLeft Task Vertex():
   maxEndTime = 0
   If left Task Vertex is not null:
      For each task Vertex in left Task Vertex:
         If task Vertex.endTime > maxEndTime:
            Set maxEndTime to task Vertex.endTime
   Return maxEndTime

Method clone():
   Create a new Task Vertex object new Task Vertex with name and duration
   Set new Task Vertex.startTime to startTime
   Set new Task Vertex.endTime to endTime
   Set new Task Vertex.left Task Vertex to null
   If left Task Vertex is not null:
      Create a new ArrayList newLeft Task Vertex
      For each task Vertex tv in left Task Vertex:
         Add tv to newLeft Task Vertex
      Set new Task Vertex.left Task Vertex to newLeft Task Vertex
   Set new Task Vertex.right Task Vertex to null
   If right Task Vertex is not null:
      Create a new ArrayList newRight Task Vertex
      For each task Vertex tv in right Task Vertex:
         Add tv to newRight Task Vertex
      Set new Task Vertex.right Task Vertex to newRight Task Vertex
   Return new Task Vertex
```

4. KESIMPULAN

Software project scheduling problem atau masalah penjadwalan proyek perangkat lunak adalah masalah tentang peangalokasian para pekerja dengan tugas-tugas yang ada sehingga dapat meminimalkan biaya dan durasi untuk keseluruhan proyek. Pada penelitian ini menggunakan pemodelan solusi dalam bentuk *gantt* diagram. Dengan struktur data binding, *gantt* diagram dapat dibangun mengikuti aturan dari *task precedence graph* sehingga urutan dan waktu pengerjaan *task* dapat menyesuaikan secara otomatis. Dari *gantt* diagram ini kemudian penjadwal dapat melihat data urutan *task*/tugas yang ada sehingga pengerjaan tugas dapat diselesaikan dengan tepat dan efisien.

REFERENSI

[1] M. Á. Vega-velázquez, A. García-nájera, and H. Cervantes, "AC," *Int. J. Prod. Econ.*, 2018, doi: 10.1016/j.ijpe.2018.04.020.
[2] P. Jalote, "Optimal Resource Allocation for the Quality Control Process," 2003.
[3] F. Chicano, F. Luna, and A. J. Nebro, "Using Multi-objective Metaheuristics to Solve the Software," pp. 1915–1922, 2011.
[4] S. Cokrowibowo, M. F. Rustan, and A. Irianti, "Pengembangan Algoritma Genetika untuk Menyelesaikan Course Scheduling Problem menggunakan Partially Mapped Crossover dan Random Pairs Mutagenesis," pp. 103–108, 2021.
[5] A. V. Rezende, L. Silva, A. Britto, and R. Amaral, "The Journal of Systems and Software Software project scheduling problem in the context of search-based software engineering : A systematic review," *J. Syst. Softw.*, vol. 155, pp. 43–56, 2019, doi: 10.1016/j.jss.2019.05.024.
[6] E. Alba and J. F. Chicano, "Software project management with GAs," vol. 177, pp. 2380–2401, 2007, doi: 10.1016/j.ins.2006.12.020.
[7] N. Nigar, "Multi-objective Dynamic Software Project Scheduling : An Evolutionary Approach for Uncertain Environments," no. December, 2020.
[8] Z. Liu *et al.*, "Data Illustrator : Augmenting Vector Design Tools with Lazy Data Binding for Expressive Visualization Authoring," pp. 1–13, 2018, doi: 10.1145/3173574.3173697.
[9] K. Wang, X. Li, L. Gao, and P. Li, "Modeling and Balancing for Green Disassembly Line Using Associated Parts Precedence Graph and Multi-objective Genetic Simulated Annealing," *Int. J. Precis. Eng. Manuf. Technol.*, no. 0123456789, 2020, doi: 10.1007/s40684-020-00259-7.
[10] H. E. Hundley, M. E. Hudson, A. D. Wasan, and T. D. Emerick, "Chronic pain clinic efficiency analysis : optimization through use of the Gantt diagram and visit diagnoses," pp. 1–8, 2019.