

Penerapan Dynamic Programming dalam Pencapaian 100% Social Link dalam Satu Playthrough Persona 3 Reload

Ramadhian Nabil Firdaus Gumay - 13524126

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jalan Ganesha 10 Bandung

E-mail: ramadhianx12@gmail.com , 13524126@std.stei.itb.ac.id

Abstrak—Makalah ini membahas penerapan algoritma Program Dinamis untuk menyelesaikan persoalan penjadwalan Social Link dalam permainan Persona 3 Reload yang dikembangkan oleh Atlus. Dalam permainan ini, pemain harus memaksimalkan seluruh 22 Social Link hingga Rank 10 dalam sekitar 300 hari permainan, di mana setiap harinya hanya tersedia slot aktivitas terbatas dengan berbagai kendala, seperti jadwal ketersediaan karakter, persyaratan statistik protagonis, dan prasyarat Social Link tertentu. Pendekatan yang digunakan adalah Program Dinamis Mundur, di mana setiap hari dihitung nilai urgency masing-masing Social Link sebagai rasio antara sesi yang masih dibutuhkan dengan jumlah hari tersedia di masa depan, kemudian Social Link dengan urgency tertinggi diprioritaskan untuk dijadwalkan. Program simulasi berbasis Python diimplementasikan dalam lima modul yang memodelkan kalender in-game, data ketersediaan Social Link, dan algoritma penjadwalan. Hasil simulasi menunjukkan bahwa algoritma berhasil menyusun jadwal sebanyak 272 sesi dalam 300 hari simulasi dan mencapai 100% Social Link. Analisis bottleneck menunjukkan bahwa Aegis (Aeon) dan Akinari Kamiki (Sun) merupakan Social Link paling kritis dengan rasio masing-masing 2.0x dan 2.8x, sesuai dengan rekomendasi panduan komunitas pemain.

Kata Kunci—Program Dinamis, Social Link, Persona 3 Reload, optimasi penjadwalan, program dinamis mundur

I. PENDAHULUAN

Dalam kehidupan sehari-hari, banyak persoalan yang pada dasarnya merupakan masalah optimasi, seperti bagaimana mengalokasikan sumber daya terbatas untuk memperoleh hasil yang maksimal. Persoalan semacam ini tidak hanya muncul di dunia nyata, tetapi juga hadir secara implisit di dalam permainan video yang memiliki sistem manajemen waktu. Salah satu contohnya adalah *Persona 3 Reload*, sebuah *role-playing game* (RPG) yang dikembangkan oleh Atlus dan dirilis pada tahun 2024. Dalam permainan ini, pemain tidak hanya bertugas menyelesaikan *dungeon* bernama Tartarus, tetapi juga harus membangun hubungan sosial dengan berbagai karakter melalui mekanisme *Social Link*.



Gambar. 1. Tampilan Menu *Social Link* Persona 3 Reload

Source: https://megatenwiki.com/wiki/Social_Link

Social Link merupakan sistem inti dalam *Persona 3 Reload* yang merepresentasikan hubungan emosional antara protagonis dengan karakter-karakter lain di sekitarnya. Terdapat 22 *Social Link* yang masing-masing terdiri dari 10 tingkatan (*rank*), sehingga total rank maksimal yang bisa dicapai adalah 220. Pencapaian rank maksimal pada semua *Social Link* dalam satu *playthrough* bukan hal yang mudah, karena setiap hari dalam permainan hanya menyediakan slot waktu terbatas, yang umumnya adalah satu sesi sore dan satu sesi malam, yang harus dialokasikan untuk berbagai aktivitas. Di samping itu, setiap *Social Link* memiliki batasan tersendiri, yaitu beberapa *Social Link* hanya tersedia pada hari atau cuaca tertentu, yang sebagian memerlukan nilai statistik protagonis, seperti *Academics*, *Charm*, dan *Courage*, yang sudah mencapai ambang batas tertentu, dan ada pula yang baru bisa diakses setelah memenuhi prasyarat cerita tertentu.

Dengan kondisi tersebut, pemain dihadapkan pada persoalan perencanaan yang cukup kompleks, yaitu bagaimana menyusun jadwal aktivitas harian selama kurang lebih 300 hari dalam permainan agar seluruh *Social Link* dapat mencapai rank 10 sebelum permainan berakhir. Jika diselesaikan hanya dengan cara coba-coba, pemain berisiko kehabisan waktu atau melewati *Social Link* tertentu. Oleh karena itu, diperlukan pendekatan yang lebih sistematis untuk memodelkan dan menyelesaikan persoalan ini.

Pendekatan yang digunakan dalam makalah ini adalah algoritma Program Dinamis untuk memodelkan persoalan penjadwalan *Social Link* tersebut. Program Dinamis adalah teknik algoritmik yang bekerja dengan memecah persoalan besar menjadi sub-persoalan yang lebih kecil, menghitung solusi optimalnya, dan menyimpan hasilnya agar tidak perlu dihitung ulang. Pendekatan ini sesuai karena persoalan penjadwalan *Social Link* memenuhi dua syarat utama Program Dinamis, yaitu *optimal substructure* dan *overlapping subproblems*.

Makalah ini memodelkan persoalan penjadwalan *Social Link* secara formal menggunakan Program Dinamis, mencakup definisi state, variabel keputusan, relasi rekurens, dan kasus basis. Untuk mendukung hal tersebut, dibuat pula program simulasi sederhana berbasis Python yang menghasilkan jadwal aktivitas harian optimal, dan hasilnya dibandingkan dengan panduan komunitas pemain Persona 3 Reload yang tersedia secara daring.

II. LANDASAN TEORI

A. Program Dinamis

Program Dinamis adalah metode pemecahan masalah optimasi yang bekerja dengan cara menguraikan solusi menjadi sekumpulan tahapan, sedemikian sehingga solusi dari suatu persoalan dapat dipandang sebagai serangkaian keputusan yang saling berkaitan. Kata “program” di sini tidak merujuk pada pemrograman komputer, melainkan berarti perencanaan, sedangkan kata “dinamis” ada karena pencarian solusinya melibatkan perhitungan menggunakan tabel yang dapat berkembang seiring berjalannya proses.

Berbeda dengan algoritma Greedy yang hanya menghasilkan satu rangkaian keputusan tunggal, Program Dinamis mempertimbangkan lebih dari satu rangkaian keputusan dan memilih yang paling optimal di antara semuanya. Hal inilah yang menjadikan Program Dinamis lebih andal untuk persoalan yang keputusan lokalnya tidak selalu menghasilkan solusi global yang optimal.

Suatu persoalan dikatakan cocok diselesaikan dengan Program Dinamis apabila memenuhi karakteristik berikut:

1. Persoalan dapat dibagi menjadi beberapa tahap, di mana pada setiap tahap hanya diambil satu keputusan.
2. Setiap tahap terdiri dari sejumlah status yang merepresentasikan kemungkinan masukan yang ada pada tahap tersebut.
3. Hasil keputusan yang diambil pada setiap tahap ditransformasikan dari status yang bersangkutan ke status berikutnya pada tahap berikutnya.
4. *Cost* pada suatu tahap meningkat secara teratur seiring bertambahnya jumlah tahapan.
5. *Cost* pada suatu tahap bergantung pada *cost* tahap-tahap sebelumnya dan *cost* dari tahap tersebut ke tahap berikutnya.
6. Solusi optimal dinyatakan dalam relasi rekurens yang menghubungkan keputusan terbaik untuk setiap status

pada tahap k dengan keputusan terbaik untuk setiap status pada tahap $k+1$.

7. Prinsip Optimalitas berlaku pada persoalan tersebut.

B. Prinsip Optimalitas dan Memoisasi

Dasar teoritis dari Program Dinamis adalah Prinsip Optimalitas yang menyatakan bahwa dalam suatu rangkaian keputusan optimal, setiap sub-rangkaian keputusan juga harus optimal terhadap kondisi yang dihasilkan oleh keputusan sebelumnya. Secara praktis, prinsip ini berarti bahwa ketika algoritma bergerak dari tahap k ke tahap $k+1$, hasil optimal dari tahap k dapat langsung digunakan tanpa menghitung ulang dari awal.

Cost pada setiap tahap dapat dinyatakan secara formal sebagai berikut:

$$Cost(k + 1) = Cost(k) + c_{\{k,k+1\}}$$

di mana $c_{\{k,k+1\}}$ adalah *cost* transisi dari tahap k ke tahap $k+1$. Dengan mengandalkan prinsip ini, Program Dinamis membangun solusi secara bertahap dari sub-persoalan yang lebih kecil menuju persoalan yang lebih besar.

Mekanisme yang memungkinkan hal tersebut berjalan efisien adalah memoisasi, yaitu teknik menyimpan hasil perhitungan setiap nilai fungsi ke dalam sebuah tabel setelah dihitung. Ketika nilai yang sama dibutuhkan kembali pada tahap berikutnya, nilai tersebut cukup diambil dari tabel tanpa harus dilakukan pemanggilan rekursif berulang. Inilah yang membedakan Program Dinamis dari pendekatan rekursif biasa dan menjadikannya jauh lebih efisien untuk persoalan yang memiliki sub-persoalan yang saling tumpang tindih.

C. Pendekatan Maju dan Mundur

Terdapat dua cara dalam melakukan perhitungan Program Dinamis:

1. Program Dinamis Maju, yaitu perhitungan yang dilakukan mulai dari tahap pertama hingga tahap akhir, yaitu dari tahap 1, 2, ..., n . Rangkaian variabel keputusan yang dihasilkan adalah $x_1, x_2, \dots, x_n$.
2. Program Dinamis Mundur, yaitu perhitungan yang dimulai dari tahap terakhir dan mundur ke tahap pertama, yaitu dari tahap $n, n-1, \dots, 1$. Rangkaian variabel keputusan yang dihasilkan adalah $x_n, x_{n-1}, \dots, x_1$.

Pemilihan antara kedua pendekatan ini umumnya disesuaikan dengan struktur persoalan. Pada makalah ini digunakan pendekatan Program Dinamis mundur, di mana perhitungan dimulai dari hari terakhir permainan dan bergerak mundur ke hari pertama. Pendekatan ini dipilih karena nilai optimal suatu hari sangat bergantung pada kondisi dan peluang yang masih tersedia di hari-hari sesudahnya.

D. Relasi Rekurens

Solusi optimal pada setiap tahap dinyatakan melalui relasi rekurens yang menghubungkan nilai optimal tahap k dengan nilai optimal tahap sebelumnya. Secara umum, untuk Program Dinamis maju, relasi rekurens dapat dituliskan sebagai:

$$f_1(s) = c_{\{x_1, s\}} \quad (\text{basis})$$

$$f_k(s) = \min / \max \{f_{\{k-1\}}(x_k) + c_{\{x_k, s\}}\} \quad (\text{rekurens})$$

di mana $f_k(s)$ adalah nilai optimal pada setiap tahap k untuk status s , x_k adalah variabel keputusan pada tahap k , dan $c_{\{x_k, s\}}$ adalah *cost* dari keputusan x_k menuju status s .

Sebagai analogi yang relevan dengan persoalan makalah ini, dapat ditinjau persoalan *Integer Knapsack* yang juga diselesaikan dengan Program Dinamis. Pada persoalan tersebut, tersedia sebuah *knapsack* berkapasitas M dan sejumlah n objek, di mana setiap objek memiliki bobot w_i dan keuntungan p_i . Tujuannya adalah memaksimalkan total keuntungan dari objek-objek yang dimasukkan ke dalam *knapsack* tanpa melebihi kapasitasnya. Relasi rekurensya adalah:

$$f_0(y) = 0, y = 0, 1, 2, \dots, M \quad (\text{basis})$$

$$f_k(y) = -\infty, y < 0 \quad (\text{basis})$$

$$f_k(y) = \max \{f_{\{k-1\}}(y), p_k + f_{\{k-1\}}(y - w_k)\}, k = 1, 2, \dots, n \quad (\text{rekurens})$$

Analogi antara *Integer Knapsack* dan persoalan penjadwalan *Social Link* cukup kuat, yaitu kapasitas *knapsack* (M) berpadanan dengan jumlah slot waktu yang tersisa, bobot objek (w_i) berpadanan dengan slot waktu yang dikonsumsi oleh satu aktivitas, dan keuntungan (p_i) berpadanan dengan kenaikan *rank Social Link* yang diperoleh. Perbedaan utamanya terletak pada dimensi state yang jauh lebih kompleks karena melibatkan banyak *Social Link* sekaligus beserta statistik protagonis.

E. Persona 3 Reload dan Sistem *Social Link*

Persona 3 Reload adalah *video game* RPG yang dikembangkan oleh Atlus dan dirilis pada Februari 2024, merupakan *remake* dari Persona 3 yang pertama kali dirilis pada tahun 2006. Permainan ini menempatkan pemain sebagai seorang protagonis pelajar SMA yang menjalani kehidupan ganda, di mana ia menjalani aktivitas sekolah dan berinteraksi dengan berbagai karakter di siang hari, dan menjelajahi *dungeon* supernatural bernama Tartarus di malam hari.

Sistem *Social Link* merupakan salah satu mekanisme inti dalam permainan ini. Setiap *Social Link* merepresentasikan tingkat kedekatan emosional antara protagonis dengan karakter tertentu dan dikaitkan dengan salah satu dari 22 Arcana Tarot yang ada di dalam permainan. Setiap *Social Link* terdiri dari 10 *rank*, dan kenaikan *rank* diperoleh dengan menghabiskan waktu bersama karakter yang bersangkutan melalui percakapan dan pilihan dialog yang tepat. Pencapaian *rank* 10 pada seluruh *Social Link* dalam satu *playthrough* merupakan tantangan tersendiri karena permainan berjalan dalam sistem kalender yang dimulai dari bulan April hingga Januari tahun berikutnya, dengan jumlah hari efektif yang terbatas.

Setiap hari dalam permainan umumnya terdiri dua slot aktivitas, yaitu sore dan malam. Pemain harus memutuskan dengan cermat aktivitas yang dilakukan pada setiap slot, seperti menghabiskan waktu bersama karakter *Social Link*,

meningkatkan statistik protagonis (*Academics*, *Charm*, *Courage*), atau menjelajahi Tartarus. Keputusan ini bersifat saling bergantung karena sejumlah *Social Link* baru dapat dinaikkan *rank*-nya setelah statistik protagonis memenuhi ambang batas minimum tertentu, dan beberapa *Social Link* hanya tersedia pada hari atau kondisi cuaca tertentu. Kompleksitas inilah yang membuat penjadwalan *Social Link* secara manual menjadi sangat rentan terhadap kesalahan alokasi waktu.

III. METODOLOGI

A. Gambaran Umum Pendekatan

Penelitian ini menggunakan pendekatan Program Dinamis Mundur untuk memodelkan persoalan penjadwalan *Social Link* dalam Persona 3 Reload. Secara umum, alur penelitian terdiri dari tiga tahap utama, yaitu pengumpulan dan pemodelan data ketersediaan *Social Link*, formulasi model Program Dinamis beserta implementasinya dalam bahasa Python, dan analisis hasil simulasi serta perbandingannya dengan panduan komunitas pemain.

B. Pengumpulan dan Pemodelan Data

Data ketersediaan seluruh 22 *Social Link* dalam Persona 3 Reload dikumpulkan dari panduan komunitas pemain yang tersedia secara daring, yaitu *Detailed Social Link Guide* yang disusun oleh HayateButler di platform Steam Community. Panduan tersebut memuat informasi lengkap mengenai jadwal ketersediaan setiap *Social Link*, persyaratan statistik protagonis yang harus dipenuhi, serta prasyarat *Social Link* lain yang perlu diselesaikan terlebih dahulu sebelum suatu *Social Link* bisa diakses.

Dari data tersebut, setiap *Social Link* dimodelkan sebagai sebuah objek dengan atribut-atribut berikut:

- Hari ketersediaan (*available_days*): hari-hari dalam seminggu di mana karakter *Social Link* bisa ditemui.
- Slot waktu (*slot*): apakah pertemuan dilakukan pada slot sore (*day*) atau malam (*night*).
- Hari pertama bisa diakses (*unlock_day*): tanggal pertama dalam kalender *in-game* di mana *Social Link* tersebut bisa dimulai.
- Persyaratan statistik (*stat_req*): nilai minimum statistik protagonis (*Academics*, *Charm*, *Courage*) yang harus terpenuhi.
- Prasyarat *Social Link* (*prereq_sl*): *Social Link* lain yang harus sudah mencapai rank tertentu sebelum *Social Link* ini bisa diakses.
- Jumlah sesi per rank (*ranks_needed*): estimasi jumlah sesi interaksi yang dibutuhkan untuk menaikkan rank dari satu level ke level berikutnya.

Kalender *in-game* dimodelkan mulai dari tanggal 7 April hingga 31 Januari tahun berikutnya, yang bila dikonversi menjadi 300 hari simulasi. Setiap hari memiliki dua slot aktivitas yang bisa diisi, yaitu slot sore dan slot malam. Selain itu, dimodelkan pula kondisi-kondisi khusus yang membatasi ketersediaan *Social Link*, antara lain minggu ujian, hari libur

nasional, serta periode musim panas yang membatasi interaksi di sekolah.

C. Definisi State dan Variabel Keputusan

Mengacu pada langkah-langkah pengembangan algoritma Program Dinamis, struktur solusi optimalnya harus didefinisikan terlebih dahulu.

State pada persoalan ini didefinisikan sebagai pasangan (hari, kondisi permainan), di mana kondisi permainan mencakup rank seluruh *Social Link* saat ini dan nilai statistik protagonis.

$$state = (hari, sl_ranks, stats)$$

di mana:

- *hari* = indeks hari dalam simulasi (0 hingga 299)
- *sl_ranks* = vektor rank seluruh 22 *Social Link* (masing-masing bernilai 0 hingga 10)
- *stats* = nilai statistik protagonis (*Academics*, *Charm*, *Courage*)

Variabel keputusan pada setiap hari adalah pilihan aktivitas yang dilakukan pada slot sore dan slot malam. Pilihan yang tersedia meliputi mengunjungi salah satu *Social Link* yang tersedia, atau melakukan aktivitas lain, seperti belajar untuk menaikkan statistik. Pada simulasi ini, fokus diberikan pada keputusan memilih *Social Link* yang akan dikunjungi.

D. Relasi Rekurens

Nilai optimal yang ingin dimaksimalkan adalah total rank *Social Link* yang bisa dicapai hingga akhir permainan. Fungsi nilai optimal didefinisikan sebagai:

$$f(hari, state) = \text{total rank Social Link yang bisa dicapai dari hari ini hingga akhir permainan, dengan state saat ini sebagai kondisi awal}$$

Relasi rekurens Program Dinamis Mundurnya adalah:

$$\begin{aligned} f(hari_{akhir}, state) &= \text{total_rank_sudah_tercapai}(state) && \text{(basis)} \\ f(hari, state) &= \max \{reward(keputusan) + f(hari + 1, state_{baru})\} && \text{(rekurens)} \end{aligned}$$

di mana *reward(keputusan)* adalah kenaikan rank yang diperoleh dari keputusan pada hari tersebut, dan *state_baru* adalah kondisi permainan setelah keputusan dieksekusi.

Karena ruang state yang sesungguhnya sangat besar, yaitu kombinasi dari 22 vektor rank (masing-masing 0-10) menghasilkan 11^{22} kemungkinan state, maka pendekatan yang digunakan adalah aproksimasi DP mundur berbasis *urgency*. Setiap hari, nilai *urgency* setiap *Social Link* dihitung sebagai berikut:

$$urgency(SL, hari) = \frac{sesi_dibutuhkan(SL)}{hari_tersisa(SL, hari)}$$

di mana *sesi_dibutuhkan* adalah jumlah sesi yang masih diperlukan untuk memaksimalkan *Social Link*, dan *hari_tersisa* adalah jumlah hari di masa depan di mana *Social Link* tersebut masih bisa dikunjungi. Nilai *urgency* ini merepresentasikan prinsip DP mundur karena keputusan hari ini ditentukan berdasarkan analisis kondisi hari-hari ke depan.

E. Implementasi Program

Program simulasi diimplementasikan dalam bahasa Python dan dibagi menjadi lima modul:

- *config.py*: mendefinisikan konstanta, kalender *in-game*, fungsi konversi tanggal, serta data hari libur dan minggu ujian.
- *social_link_data.py*: menyimpan data ketersediaan seluruh 22 *Social Link*, termasuk jadwal, persyaratan statistik, dan estimasi jumlah sesi per rank.
- *scheduler.py*: mengimplementasikan logika DP, fungsi pengecekan ketersediaan *Social Link* per hari, perhitungan *urgency*, dan algoritma penjadwalan utama.
- *output.py*: menangani seluruh fungsi tampilan hasil simulasi.
- *main.py*: menjalankan seluruh proses secara berurutan.

Algoritma penjadwalan pada *scheduler.py* bekerja dengan cara mengiterasi setiap hari dari hari pertama hingga hari terakhir. Pada setiap hari, program mengumpulkan semua *Social Link* yang tersedia berdasarkan kondisi state saat itu, menghitung *urgency* masing-masing, lalu memilih *Social Link* dengan *urgency* tertinggi untuk dimasukkan ke dalam jadwal. Proses ini dilakukan secara terpisah untuk slot sore dan slot malam.

Fungsi cek ketersediaan *is_sl_available()* memverifikasi enam kondisi secara berurutan, yaitu apakah hari ini sudah melewati tanggal unlock, apakah belum melewati tanggal akhir, apakah rank belum maksimal, apakah statistik protagonis sudah memenuhi syarat, apakah prasyarat *Social Link* lain sudah terpenuhi, dan apakah hari ini termasuk hari yang dijadwalkan untuk *Social Link* tersebut.

F. Analisis Bottleneck

Sebelum menjalankan simulasi penjadwalan, program terlebih dahulu menjalankan analisis *bottleneck* menggunakan komponen DP mundur. Analisis ini menghitung jumlah hari tersedia dan jumlah sesi yang dibutuhkan untuk setiap *Social Link*, lalu mengurutkannya berdasarkan rasio keduanya. *Social Link* dengan rasio paling kecil berarti paling kritis karena memiliki sedikit waktu tersedia relatif terhadap banyaknya sesi waktu yang dibutuhkan. Hasil analisis ini digunakan sebagai dasar untuk memverifikasi bahwa algoritma penjadwalan memprioritaskan *Social Link* yang tepat.

IV. HASIL DAN ANALISIS

A. Hasil Analisis Bottleneck

Sebelum menjalankan simulasi penjadwalan, program terlebih dahulu melakukan analisis *bottleneck* menggunakan komponen DP mundur untuk mengidentifikasi *Social Link* mana yang paling kritis dari sisi waktu. Hasil analisis ditampilkan pada Tabel I.

TABLE I. HASIL ANALISIS BOTTLENECK

No	Social Link	Hari Tersedia	Sesi Dibutuhkan	Rasio
1	Aigis (Aeon)	20	10	2.0x
2	Akinari Kamiki (Sun)	25	9	2.8x
3	Maya (Hermit)	49	15	3.3x
4	Mitsuru Kirijo (Empress)	50	16	3.1x
5	Yuko Nishiwaki (Strength)	69	18	3.8x
6	President Tanaka (Devil)	75	9	8.3x
7	Mamoru Hayase (Star)	77	16	4.8x
8	Keisuke Hiraga (Fortune)	85	15	5.7x
9	Fuuka Yamagishi (Priestess)	86	16	5.4x
10	Yukari Takeba (Lovers)	96	15	6.4x
11	Bebe (Temperance)	99	15	6.6x
12	Chihiro Fushimi (Justice)	102	17	6.0x
13	Hidetoshi Odagiri (Emperor)	103	12	8.6x
14	Kenji Tomochika (Magician)	107	13	8.2x
15	Maiko Oohashi (Hanged-Man)	116	16	7.2x
16	Kazushi Miyamoto (Chariot)	142	17	8.4x
17	Mutatsu (Tower)	145	16	9.1x
18	Bunkichi & Mitsuko (Hierophant)	241	18	13.4x
19	Nozomi Suemitsu (Moon)	278	9	30.9x

Dari hasil analisis tersebut, terdapat beberapa temuan. Aigis (Arcana Aeon) merupakan *Social Link* paling kritis dengan rasio hanya 2.0x, artinya jumlah hari tersedia hanya dua kali lipat dari sesi yang dibutuhkan. Hal ini terjadi karena Aigis baru bisa diakses mulai tanggal 8 Januari dan hanya tersedia hingga 30 Januari, sehingga total jendela waktu yang dimiliki hanya sekitar 20 hari. Dengan 10 sesi yang harus diselesaikan dalam rentang waktu tersebut, hampir tidak ada toleransi untuk melewatkan satu hari pun.

Social Link paling kritis berikutnya adalah Akinari Kamiki (Arcana Sun) dengan rasio 2.8x. Sun hanya tersedia pada hari Minggu dan baru bisa diakses mulai Agustus, sehingga total hari yang tersedia sangat terbatas. Di sisi lain, Nozomi Suemitsu (Arcana Moon) memiliki rasio paling longgar yaitu 30.9x karena tersedia setiap hari dalam seminggu sejak akhir April, sehingga tidak menjadi kendala sama sekali dalam penjadwalan.

Temuan ini menunjukkan bahwa tanpa pendekatan yang sistematis, pemain sangat mudah gagal memaksimalkan Aigis dan Akinari karena tidak menyadari betapa sempitnya jendela waktu yang tersedia untuk kedua karakter tersebut.

B. Hasil Simulasi Penjadwalan

Setelah analisis *bottleneck*, program menjalankan simulasi penjadwalan menggunakan algoritma DP mundur berbasis *urgency*. Program berhasil menyusun jadwal lengkap sebanyak 272 sesi dalam 300 hari simulasi, dan berhasil memaksimalkan seluruh 22 *Social Link* hingga Rank 10.

RINGKASAN HASIL SIMULASI			
Social Link selesai (Rank 10) : 22/22			
Total rank tercapai : 220/220			
Persentase : 100.0%			

Gambar. 2. Ringkasan Hasil Simulasi

Tabel II menampilkan 30 sesi pertama dari jadwal yang dihasilkan oleh program sebagai ilustrasi bagaimana algoritma menyusun urutan kunjungan.

TABLE II. SAMPEL JADWAL 30 SESI PERTAMA

Tanggal	Slot	Social Link	Rank
April 23 (Kam)	Sore	Kenji Tomochika (Magician)	0 → 2
Apr 24 (Jum)	Sore	Kazushi Miyamoto (Chariot)	0 → 2
Apr 25 (Sab)	Sore	Yuko Nishiwaki (Strength)	0 → 1
Apr 26 (Min)	Sore	Bunkichi & Mitsuko (Hierophant)	0 → 1
Apr 27 (Sen)	Sore	Hidetoshi Odagiri (Emperor)	0 → 2
Apr 28 (Sel)	Sore	Kazushi Miyamoto (Chariot)	2 → 2
Apr 29 (Rab)	Sore	Maya (Hermit)	0 → 2
Apr 30 (Kam)	Sore	Kazushi Miyamoto (Chariot)	2 → 3
Mei 1 (Jum)	Sore	Kenji Tomochika (Magician)	2 → 2
Mei 2 (Sab)	Sore	Yuko Nishiwaki (Strength)	1 → 2
Mei 3 (Min)	Sore	Maya (Hermit)	2 → 2
Mei 4 (Sen)	Sore	Maya (Hermit)	2 → 3
Mei 5 (Sel)	Sore	Maya (Hermit)	3 → 4
Mei 6 (Rab)	Sore	Yuko Nishiwaki (Strength)	2 → 2
Mei 7 (Kam)	Sore	Chihiro Fushimi (Justice)	0 → 1
Mei 8 (Jum)	Sore	Kenji Tomochika (Magician)	2 → 3
Mei 9 (Sab)	Sore	Yuko Nishiwaki (Strength)	2 → 3
Mei 10 (Min)	Sore	Maya (Hermit)	4 → 4
Mei 11 (Sen)	Sore	Maiko Oohashi (Hanged-Man)	0 → 2
Mei 12 (Sel)	Sore	Chihiro Fushimi (Justice)	1 → 2
Mei 13 (Rab)	Sore	Yuko Nishiwaki (Strength)	3 → 3
Mei 14 (Kam)	Sore	Chihiro Fushimi (Justice)	2 → 2
Mei 15 (Jum)	Sore	Hidetoshi Odagiri (Emperor)	2 → 3
Mei 16 (Sab)	Sore	Yuko Nishiwaki (Strength)	3 → 4
Mei 16 (Sab)	Malam	President Tanaka (Devil)	0 → 2
Mei 17 (Min)	Sore	Maya (Hermit)	4 → 5
Mei 18 (Sen)	Sore	Maiko Oohashi (Hanged-Man)	2 → 2
Mei 19 (Sel)	Sore	Chihiro Fushimi (Justice)	2 → 3
Mei 19 (Sel)	Malam	President Tanaka (Devil)	2 → 3
Mei 20 (Rab)	Sore	Maiko Oohashi (Hanged-Man)	2 → 3

Dari sampel tersebut, terlihat bahwa pada hari-hari pertama bulan April, algoritma langsung memprioritaskan beberapa *Social Link* sekaligus secara bersamaan, yaitu Kenji Tomochika (Magician), Kazushi Miyamoto (Chariot), dan Yuko Nishiwaki (Strength). Ketiga *Social Link* ini diprioritaskan di awal karena nilai *urgency*-nya sudah cukup tinggi sejak hari pertama, karena tersedia di hari-hari sekolah saja tetapi membutuhkan banyak sesi untuk mencapai Rank 10. Di samping itu, pada tanggal 29 April yang merupakan hari libur *Showa Day*, algoritma langsung menjadwalkan kunjungan ke Maya (Hermit) karena hari tersebut adalah salah satu dari sedikit hari di mana Hermit bisa ditemui.

SELURUH JADWAL (TOTAL 272 SESI)			
Tanggal	Slot	Social Link	Rank
Apr 23 (Kam)	Sore	Kenji Tomochika (Magician)	0 → 2
Apr 24 (Jum)	Sore	Kazushi Miyamoto (Chariot)	0 → 2
Apr 25 (Sab)	Sore	Yuko Nishiwaki (Strength)	0 → 1
Apr 26 (Min)	Sore	Bunkichi & Mitsuko (Hierophant)	0 → 1
Apr 27 (Sen)	Sore	Hidetoshi Odagiri (Emperor)	0 → 2
Apr 28 (Sel)	Sore	Kazushi Miyamoto (Chariot)	2 → 2
Apr 29 (Rab)	Sore	Maya (Hermit)	0 → 2
Apr 30 (Kam)	Sore	Kazushi Miyamoto (Chariot)	2 → 3
Mei 1 (Jum)	Sore	Kenji Tomochika (Magician)	2 → 2
Mei 2 (Sab)	Sore	Yuko Nishiwaki (Strength)	1 → 2
Mei 3 (Min)	Sore	Maya (Hermit)	2 → 2
Mei 4 (Sen)	Sore	Maya (Hermit)	2 → 3
Mei 5 (Sel)	Sore	Maya (Hermit)	3 → 4
Mei 6 (Rab)	Sore	Yuko Nishiwaki (Strength)	2 → 2
Mei 7 (Kam)	Sore	Chihiro Fushimi (Justice)	0 → 1
Mei 8 (Jum)	Sore	Kenji Tomochika (Magician)	2 → 3
Mei 9 (Sab)	Sore	Yuko Nishiwaki (Strength)	2 → 3
Mei 10 (Min)	Sore	Maya (Hermit)	4 → 4
Mei 11 (Sen)	Sore	Maiko Oohashi (Hanged-Man)	0 → 2
Mei 12 (Sel)	Sore	Chihiro Fushimi (Justice)	1 → 2
Mei 13 (Rab)	Sore	Yuko Nishiwaki (Strength)	3 → 3
Mei 14 (Kam)	Sore	Chihiro Fushimi (Justice)	2 → 2
Mei 15 (Jum)	Sore	Hidetoshi Odagiri (Emperor)	2 → 3
Mei 16 (Sab)	Sore	Yuko Nishiwaki (Strength)	3 → 4
Mei 16 (Sab)	Malam	President Tanaka (Devil)	0 → 2
Mei 17 (Min)	Sore	Maya (Hermit)	4 → 5
Mei 18 (Sen)	Sore	Maiko Oohashi (Hanged-Man)	2 → 2
Mei 19 (Sel)	Sore	Chihiro Fushimi (Justice)	2 → 3
Mei 19 (Sel)	Malam	President Tanaka (Devil)	2 → 3
Mei 20 (Rab)	Sore	Maiko Oohashi (Hanged-Man)	2 → 3

Gambar. 3. Sampel Jadwal di Terminal

Tabel III menampilkan rank akhir seluruh *Social Link* setelah simulasi selesai.

TABLE III. RANK AKHIR SEMUA SOCIAL LINK

Arcana	Karakter	Status
Fool	S.E.E.S	MAX
Magician	Kenji Tomochika	MAX
Priestess	Fuuka Yamagishi	MAX
Empress	Mitsuru Kirijo	MAX
Emperor	Hidetoshi Odagiri	MAX
Hierophant	Bunkichi & Mitsuko	MAX
Lovers	Yukari Takeba	MAX
Chariot	Kazushi Miyamoto	MAX
Justice	Chihiro Fushimi	MAX
Hermit	Maya	MAX
Fortune	Keisuke Hiraga	MAX
Strength	Yuko Nishiwaki	MAX
Hanged-Man	Maiko Oohashi	MAX
Death	Pharos	MAX
Temperance	Bebe	MAX
Devil	President Tanaka	MAX
Tower	Mutatsu	MAX

Arcana	Karakter	Status
Moon	Nozomi Suemitsu	MAX
Sun	Akinari Kamiki	MAX
Judgement	Nyx Annihilation Team	MAX
Aeon	Aigis	MAX

C. Analisis Efisiensi Penjadwalan

Dari total 300 hari simulasi, algoritma menghasilkan 272 sesi terjadwal. Artinya terdapat sekitar 28 hari di mana tidak ada *Social Link* yang dijadwalkan, baik karena semua *Social Link* yang tersedia pada hari tersebut sudah mencapai Rank 10, maupun karena tidak ada *Social Link* yang memenuhi seluruh syarat ketersediaan pada hari itu.

Dari 272 sesi tersebut, 174 sesi menghasilkan kenaikan rank secara langsung, sedangkan 98 sesi merupakan sesi lanjutan untuk *Social Link* yang membutuhkan lebih dari satu sesi per rank. Kondisi ini terjadi karena beberapa *Social Link* seperti Chariot, Strength, dan Hierophant memiliki nilai *rank_needed* sebesar 2 untuk beberapa rank tertentu, artinya dibutuhkan dua kali kunjungan sebelum rank naik satu tingkat.

Dari sisi distribusi waktu, sebagian besar sesi terjadwal pada slot sore karena mayoritas *Social Link* dalam Persona 3 Reload memang hanya tersedia pada siang hari. Hanya dua *Social Link* yang menggunakan slot malam, yaitu President Tanaka (Devil) dan Mutatsu (Tower), sehingga slot malam relatif jarang terisi.

D. Perbandingan dengan Panduan Komunitas

Untuk mengontekstualisasi hasil yang diperoleh, jadwal yang dihasilkan oleh algoritma dibandingkan dengan panduan komunitas pemain Persona 3 Reload yang tersedia secara daring. Perbandingan ini dilakukan secara kualitatif berdasarkan urutan prioritas *Social Link* yang disarankan.

Secara umum, urutan prioritas yang dihasilkan oleh algoritma sejalan dengan rekomendasi yang ada di panduan komunitas. Panduan komunitas secara eksplisit menyarankan pemain untuk segera mengurus Aigis (Aeon) begitu tersedia di awal Januari, serta tidak menunda-nunda Sun karena ketersediaannya yang terbatas hanya di hari Minggu. Hal ini sesuai dengan hasil analisis *bottleneck* yang menempatkan keduanya sebagai *Social Link* paling kritis.

Perbedaan yang ditemukan terletak pada fleksibilitas pendekatan. Panduan komunitas cenderung memberikan rekomendasi yang *fixed* untuk setiap tanggal tertentu, sedangkan algoritma DP yang diimplementasikan bersifat adaptif, keputusan setiap harinya dihitung ulang berdasarkan kondisi state saat itu. Hal ini membuat jadwal yang dihasilkan algoritma lebih fleksibel terhadap perubahan kondisi, misalnya jika pemain terlambat memulai suatu *Social Link* karena alasan tertentu.

Selain itu, algoritma berhasil menyelesaikan seluruh 22 *Social Link* dalam 272 sesi, yang berarti masih terdapat cukup banyak slot waktu tersisa yang bisa digunakan pemain untuk aktivitas lain seperti menjelajahi Tartarus atau menaikkan

statistik protagonis. Ini menunjukkan bahwa dengan penjadwalan yang tepat, pencapaian 100% *Social Link* dalam satu *playthrough* bukanlah hal yang mustahil tanpa harus mengorbankan aspek permainan lainnya.

E. Keterbatasan Simulasi

Terdapat beberapa hal yang perlu diperhatikan terkait keterbatasan dari simulasi yang dijalankan. Pertama, jumlah sesi per rank (*ranks_needed*) yang digunakan dalam simulasi merupakan estimasi minimum berdasarkan panduan komunitas dengan asumsi pemain selalu memilih dialog yang optimal dan menggunakan Persona dengan Arcana yang sesuai. Dalam kondisi permainan nyata tanpa bonus tersebut, jumlah sesi yang dibutuhkan bisa lebih banyak.

Kedua, simulasi ini tidak memodelkan *forced events* atau kejadian cerita tertentu yang menghalangi aktivitas bebas pada hari-hari tertentu, seperti malam bulan purnama atau periode menjelang pertarungan bos. Pada permainan sesungguhnya, terdapat sejumlah hari di mana pemain tidak bisa memilih aktivitas secara bebas.

Ketiga, kenaikan statistik protagonis (*Academics*, *Charm*, *Courage*) dimodelkan secara sederhana menggunakan jadwal yang sudah ditentukan di awal, bukan dihitung secara dinamis berdasarkan aktivitas harian. Dalam permainan sesungguhnya, statistik naik tergantung dari aktivitas yang dipilih pemain di luar kunjungan *Social Link*.

V. KESIMPULAN

Dari hasil simulasi yang dijalankan, algoritma berhasil menyusun jadwal sebanyak 272 sesi dalam 300 hari simulasi dan berhasil memaksimalkan 22 *Social Link* hingga Rank 10, yang berarti target 100% *Social Link* dalam satu *playthrough* tercapai. Hasil analisis *bottleneck* menunjukkan bahwa Aegis (Arcana Aeon) dan Akinari Kamiki (Arcana Sun) merupakan *Social Link* paling kritis dengan rasio masing-masing sebesar 2.0x dan 2.8x, sehingga keduanya harus diprioritaskan sejak awal tersedia. Temuan ini sesuai dengan rekomendasi panduan komunitas yang juga menekankan pentingnya kedua *Social Link* tersebut. Dibandingkan dengan panduan komunitas yang bersifat tetap untuk setiap tanggal tertentu, pendekatan algoritma yang diimplementasikan bersifat lebih adaptif karena keputusan setiap harinya dihitung ulang berdasarkan kondisi *state* saat itu.

Terdapat beberapa keterbatasan yang perlu diakui, yaitu simulasi ini menggunakan estimasi minimum jumlah sesi per rank, tidak memodelkan *forced events* dari cerita permainan, dan menghitung kenaikan statistik protagonis secara sederhana. Untuk penelitian selanjutnya, model ini dapat disempurnakan dengan memasukkan data *forced events* secara lengkap, memodelkan kenaikan statistik secara dinamis, serta mengeksplorasi penerapan DP penuh dengan kompresi *state*

untuk mengatasi persoalan ledakan dimensi ruang *state* yang pada penelitian ini mencapai 11^{22} kemungkinan.

REPOSITORY LINK AT GITHUB

<https://github.com/Rmadhian/p3r-social-link-dp-scheduler>

ACKNOWLEDGMENT

Puji dan syukur penulis panjatkan ke hadirat Allah SWT. atas rahmat dan karunia-Nya sehingga makalah ini dapat diselesaikan dengan baik. Penulis menyampaikan terima kasih kepada Dr. Ir. Rinaldi Munir, M.T., Dr. techn. Muhammad Zuhri Catur Candra, S.T., M.T., Tricya Esterina Widagdo, S.T., M.Sc., dan Dr. Fazat Nur Azizah, S.T., M.Sc., selaku dosen mata kuliah IF2211 Strategi Algoritma, atas bimbingan dan ilmu yang diberikan selama semester ini sehingga penulis dapat menyelesaikan penelitian ini.

REFERENCES

- [1] Atlus, *Persona 3 Reload*, Sega, Feb. 2024.
- [2] R. Munir, "Program Dinamis (Dynamic Programming) - Bagian 1," Bahan Kuliah IF2211 Strategi Algoritma, Institut Teknologi Bandung, 2026. [Online]. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2025-2026/25-Program-Dinamis-\(2026\)-Bagian1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2025-2026/25-Program-Dinamis-(2026)-Bagian1.pdf)
- [3] R. Munir, "Program Dinamis (Dynamic Programming) - Bagian 2," Bahan Kuliah IF2211 Strategi Algoritma, Institut Teknologi Bandung, 2026. [Online]. Available: [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2025-2026/26-Program-Dinamis-\(2026\)-Bagian2.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2025-2026/26-Program-Dinamis-(2026)-Bagian2.pdf)
- [4] HayateButler, "Detailed Social Link Guide - Persona 3 Reload," *Steam Community*, 2024. [Online]. Available: <https://steamcommunity.com/sharedfiles/filedetails/?id=3152126713>

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Juni 2026



Ramadhian Nabil Firdaus Gumay, 13524126