



Sistem Rekomendasi Komponen Rakit PC dengan Algoritma *Greedy* dan *Rule-Based Filtering*

Samuel Yahya¹, Daniel Prasetyo Dodi Darmawan^{2*}, Gabriel Putra³,
Timothy Christian⁴, Vitri Tundjung Sari⁵

¹⁻⁵Faculty of Computer Science, Universitas Esa Unggul, Indonesia

*Penulis Korespondensi: danielprasetyo053@gmail.com

Abstract. Various computer components are developing rapidly due to the increasing need for users to assemble PCs or personal computers to suit their work needs, such as 3D work, graphic design, video editing, gaming, and others. However, variety, compatibility, and price often make it difficult for many users to determine the optimal combination that is comparable to the costs incurred. This research aims to build a simple PC component recommendation system that can provide the best component suggestions according to the user's budget. This system applies the Greedy Algorithm with additional context to select components. Selection is prioritized based on CPU Mark and G3D Mark scores. Strict rule-based filtering is applied at the beginning to filter out low-quality components, ensuring that only modern and reliable components are included. The system was developed using a PC component dataset obtained from PCPartPicker, which includes core PC components such as CPU, RAM, storage, case, and PSU. The test results show that a simple recommendation system built with the Greedy Algorithm and Rule-Based Filtering can produce an efficient and budget-friendly PC assembly. The combination of the two methods results in a recommendation system that is fast, accurate, and relevant to the results produced.

Keywords: Computer Components; Computers; Greedy Algorithm; Recommendation System; Rule-Based Filtering.

Abstrak. Beragam komponen komputer berkembang dengan pesat karena meningkatnya kebutuhan pengguna merakit PC atau personal computer untuk sesuai kebutuhan pekerjaan seperti pekerjaan 3D, desain grafis, video editing, gaming, dan lain-lain. Namun variasi, kompatibilitas, dan harga sering membuat banyak pengguna kesulitan menentukan kombinasi yang optimal dan sebanding dengan biaya yang dikeluarkan. Penelitian ini bertujuan membangun sistem rekomendasi komponen rakit pc sederhana yang dapat memberikan saran komponen terbaik sesuai anggaran pengguna. Sistem ini menerapkan Algoritma Greedy dengan tambahan konteks untuk memilih komponen. Pemilihan diprioritaskan berdasarkan skor CPU Mark dan G3D Mark. Rule-based filtering yang ketat diterapkan di awal untuk menyaring komponen berkualitas rendah, memastikan hanya ada komponen yang modern dan andal. Sistem dikembangkan dengan memanfaatkan dataset komponen PC yang didapat dari PCPartPicker yang mencakup komponen inti dari PC seperti CPU, RAM, storage, case, dan PSU. Hasil pengujian menunjukkan sistem rekomendasi sederhana yang dibangun dengan algoritma Greedy dan Rule-Based Filtering dapat menghasilkan rakitan PC yang efisien dan sesuai anggaran. Kombinasi antara kedua metode menghasilkan sistem rekomendasi yang cepat, akurat, dan relevan dengan hasil yang dikeluarkan.

Kata kunci: Algoritma Greedy; Komponen Komputer; Komputer; Rule-Based Filtering; Sistem Rekomendasi.

1. LATAR BELAKANG

PC atau Personal Computer merupakan alat keseharian mayoritas orang-orang di dunia bekerja pada saat ini. Komputer adalah alat yang dipakai untuk mengolah berbagai data sesuai dengan prosedur yang telah ditentukan atau dirumuskan oleh pengguna (Utomo & Mustafidah, 2016). Komputer juga digunakan untuk media mendapatkan berbagai informasi, pendidikan, hiburan, dan media untuk membuat desain grafis, dan animasi khususnya dalam bidang industri kreatif. Dari semua kebutuhan itu komputer sudah menjadi kebutuhan primer dan tidak tergantikan oleh banyak orang (Saputra, 2024). Tentunya kebutuhan performa komputer ditentukan dari pengguna itu sendiri. Sebagai contoh, keperluan untuk animasi 3D memerlukan performa komputer yang lebih tinggi daripada performa komputer yang cukup untuk bermain

game online saja. Begitupun sebaliknya jika seseorang hanya melakukan kegiatan produktivitas biasa seperti mengetik dokumen atau melakukan pencatatan, tidak diperlukan untuk membeli atau merakit komputer dengan performa tinggi.

Komputer memiliki banyak tipe ataupun jenis mulai tipe performa tipe standar, menengah, dan berteknologi atau berperforma tinggi (Ginting, 2022). Terdapat juga komputer yang sudah dirakit oleh sebuah brand atau vendor biasanya disebut built-up. Komputer yang telah dirakit oleh brand biasanya hanya mencakup sebuah rentang harga tanpa mempedulikan alokasi harga komponen yang ada. Sebagai contoh, komputer untuk melakukan berbagai pekerjaan multitasking seperti melakukan banyak containerization dengan docker untuk melakukan pengembangan perangkat lunak membutuhkan komponen CPU yang memiliki banyak core supaya tidak menghambat kinerja multitasking (Preeth et al., 2015). Maka dari itu, alokasi budget dapat lebih besar untuk komponen CPU, yang dimana hal ini biasanya tidak dilakukan oleh vendor atau suatu jenis brand. Selain dari keuntungan tersebut, terdapat banyak pilihan komponen dari banyak lain brand yang memiliki fitur lebih atau kurang tergantung dari budget yang dimiliki oleh calon perakit PC. Komponen bisa dikurangi ataupun ditambahkan jika memang dibutuhkan, hal ini biasanya tidak dapat dilakukan oleh vendor atau brand seperti HP dan Dell (Ardhana & Mulyodiputro, 2023).

Banyaknya komponen komputer dengan harga yang berbeda-beda pastilah timbul kebingungan khususnya bagi pengguna yang awam merakit komputer. Penelitian ini dibuat untuk mengimplementasikan algoritma sederhana seperti Greedy dan Rule-based untuk membuat sistem rekomendasi komponen komputer yang tidak hanya cepat namun juga efektif memberikan hasil rekomendasi. Algoritma Greedy karena kesederhanaan logikanya dan dalam kondisi praktis, lebih efisien dan memiliki performa yang baik (Pokutta et al., 2020). Dengan penggunaan algoritma ini, diharapkan sistem dapat membuat rekomendasi yang cukup baik.

2. KAJIAN TEORITIS

Algoritma Greedy

Algoritma Greedy adalah metode algoritma paling populer untuk menyelesaikan berbagai permasalahan optimasi. Permasalahan optimasi adalah sebuah permasalahan yang dimana membutuhkan solusi optimal atau solusi terbaik. Algoritma Greedy sangat sederhana maka dari itu implementasinya fleksibel dan dapat digunakan diberbagai kasus dengan hasil yang cukup memuaskan (Roihan et al., 2022). Konsep dari algoritma Greedy adalah mirip seperti algoritma pencarian yang lain yaitu memilih solusi terbaik secara lokal terlebih dahulu dan berharap solusi tersebut menghasilkan solusi terbaik secara global (Siregar et al., 2025).

Walaupun algoritma Greedy memiliki banyak keunggulan, namun kelemahan algoritma Greedy rentan membuat pilihan kurang baik jika banyak alternatif sehingga membuat keputusan yang kurang baik secara global karena cara kerja algoritma Greedy yang memilih solusi terbaik secara lokal (Dima et al., 2025; Ndruru & Lubis, 2025). Algoritma Greedy bekerja dengan prinsip memilih solusi lokal terbaik pada setiap tahap dengan harapan menghasilkan solusi global yang optimal atau mendekati optimal, terutama pada permasalahan optimasi yang memiliki struktur optimal substructure dan greedy choice property. Dalam konteks sistem rekomendasi rakit komputer, pendekatan ini tidak hanya mempertimbangkan nilai tertinggi secara absolut seperti harga, tetapi juga mempertimbangkan prioritas dan kendala seperti anggaran, kompatibilitas, dan kebutuhan performa, sehingga pemilihan komponen inti seperti CPU dan GPU dilakukan terlebih dahulu berdasarkan kontribusi performa terbesar terhadap sistem. Selanjutnya, komponen pendukung dipilih berdasarkan aturan spesifik seperti kompatibilitas motherboard, kecepatan RAM, dan efisiensi biaya, yang sejalan dengan konsep sistem rekomendasi berbasis aturan (rule-based recommendation) yang menggunakan pengetahuan domain dan constraint untuk menghasilkan rekomendasi yang relevan dan kontekstual bagi pengguna (Cormen et al., 2009; Ricci et al., 2015; Russell & Norvig, 2021).

Rule-based Filtering

Rule-base Filtering adalah metode penyimpanan data yang menggunakan logika if-then untuk menentukan apakah suatu item memenuhi kriteria tertentu. Rule-based Filtering sangat sederhana dan ringan untuk dijalankan sehingga cocok untuk pembangunan sistem rekomendasi. Dengan algoritma ini, maka setiap komponen memiliki kecocokan dan saling kompatibel (Maulana, 2025).

Dalam penelitian ini, aturan yang digunakan mencakup beberapa aspek kompatibilitasnya, diantaranya:

- 1) CPU dan Motherboard: kesesuaian tipe socket (misalnya LGA1200, AM4)
- 2) Motherboard dan RAM: kesesuaian tipe DDR (DDR4/DDR5) dan kapasitas maksimum yang didukung.
- 3) Motherboard dan Casing: kesesuaian ukuran form factor (ATX, mATX, ITX).
- 4) PSU dan GPU: kecukupan daya (wattage) dan konektor daya yang sesuai.
- 5) Storage dan Motherboard: kesesuaian antarmuka (SATA atau NVMe).

Tahap ini dijalankan sesudah seleksi Greedy, sehingga hanya komponen yang lulus uji kompatibilitas yang akan direkomendasikan kepada pengguna. Dengan kombinasi algoritma Greedy dan Rule-based, sistem dapat menghasilkan rekomendasi rakitan PC yang kompatibel, efisien, dan sesuai dengan kebutuhan pengguna.

3. METODE PENELITIAN

Dataset

Data yang digunakan dalam penelitian ini diperoleh dari situs PCPartPicker, sebuah platform penyedia, informasi lengkap mengenai spesifikasi, harga, serta kompatibilitas antar komponen komputer. Proses pengumpulan data dilakukan dengan metode *web scraping*, kemudian disimpan dalam format CSV (*Comma-Separated Values*) agar mudah diolah lebih lanjut. Link menuju dataset: <https://github.com/docyx/pc-part-dataset/tree/main/data/csv>. Selain dari dataset tersebut, skor *cpu* dan *gpu* diambil dari website cpubenchmark.net dan videocardbenchmark.net menggunakan alat ekstraksi HTML.

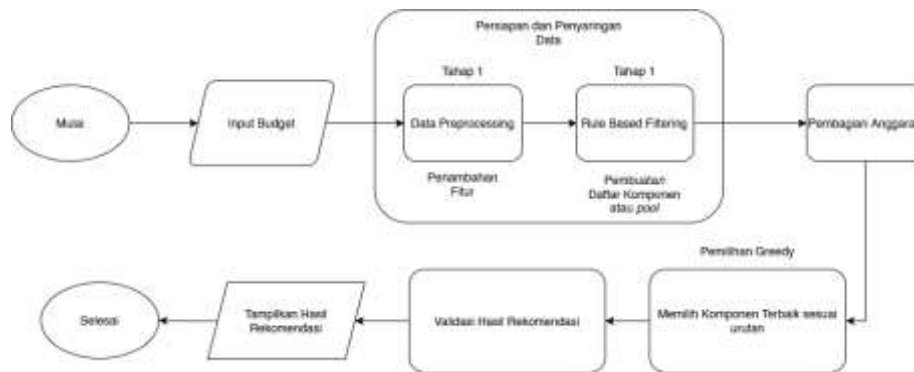
Dataset penelitian diperoleh dari PCPartPicker dalam format CSV yang mencakup berbagai komponen PC, seperti CPU, GPU, motherboard, RAM, storage, PSU, dan casing. Dataset ini kemudian diperkaya dengan data benchmark CPU Mark dan G3D Mark yang diambil dari situs benchmark terkemuka. Penambahan atribut kompatibilitas dan klasifikasi komponen dilakukan untuk mendukung proses filtering dan seleksi.

Desain Sistem

Sistem rekomendasi dirancang melalui beberapa tahap utama, yaitu input anggaran oleh pengguna, preprocessing data, seleksi komponen menggunakan algoritma Greedy, penyaringan kompatibilitas menggunakan Rule-Based Filtering, serta penyusunan hasil rekomendasi akhir berupa daftar komponen dan total biaya. Implementasi sistem dilakukan menggunakan bahasa pemrograman Python dengan bantuan pustaka *pandas* untuk pengolahan data.

Alat Implementasi

Dalam tahap implementasi, penelitian ini menggunakan beberapa alat dan pustaka (*library*) untuk mendukung perancangan sistem, yaitu bahasa pemrograman *python* sebagai bahasa utama pengembangan implementasi sistem rekomendasi. Lingkungan pengembangan atau *code editor* yang digunakan adalah *Visual Studio Code* atau *VSCode* untuk pengembangan yang mudah, efisien, dan cepat. Library yang digunakan adalah *pandas*, *pandas* adalah *library* untuk manipulasi dan analisis data. *Pandas* pada penelitian ini digunakan untuk membaca dan menyimpan data eksternal yang disimpan dalam file *csv*. Adapun alur kerja pada penelitian ini sebagai berikut.



Gambar 1. Diagram Alur Sistem Rekomendasi Rakit PC.

4. HASIL DAN PEMBAHASAN

Hasil

Pertama, Pengguna memasukkan anggaran yang dimiliki untuk merakit PC dalam satuan Rupiah sebagai batas maksimum biaya yang dapat digunakan untuk merakit PC. Kedua, setiap dataset yang digunakan harus difilter atau diberikan suatu aturan untuk menjaga relevansi dari rakitan komputer seperti contoh cpu sebisa mungkin adalah generasi terbaru dan bukan versi lama. Beberapa penambahan kolom atau informasi diperlukan untuk membuat sistem lebih canggih dan manusiawi.

- 1) Menambahkan kolom socket dan *tdp_max* pada CPU untuk memastikan socket CPU sesuai dengan model yang ada berdasarkan generasinya dan TDP maksimal yang sekiranya diperlukan untuk menjaga kestabilan komputer dan pemilihan PSU yang lebih optimal. *Tdp_max* diperlukan sehingga sistem tidak memilih PSU yang memiliki daya yang sama, sekiranya PSU memiliki daya yang lebih besar untuk mengakomodasikan kestabilan komputer.
- 2) Menambahkan kolom *tdp* pada GPU untuk informasi daya yang minimal yang diperlukan. Informasi ini didapat dari spesifikasi chipset GPU tersebut menggunakan *hardcode*
- 3) Menambahkan kolom *ram type* pada *motherboard* untuk mengetahui tipe RAM yang berlaku pada *motherboard* tersebut. Penambahan kolom tambahan yaitu *tier* atau tingkat untuk mengetahui tingkat motherboard seperti seri Z, seri B, dan A sehingga sistem dapat memilih tingkat motherboard sesuai dengan jangkauan input budget yang ada. Penentuan tingkat sesuai seri tersebut dilakukan secara *hardcode*. Kolom *tier* digunakan untuk membuat daftar *motherboard* yang sesuai untuk beberapa kondisi budget dan pemilihan komponen sementara. Budget diatas atau sama dengan 25 juta

maka akan mendapatkan *motherboard* tingkat 3 dan budget diatas atau sama dengan 12 juta mendapatkan *motherboard* tingkat 2.

- 4) Menambahkan kolom *type* pada RAM untuk mengetahui tipe apa berdasarkan speed yang terdapat pada dataset. DDR4 berada pada jangkauan nilai 2133 sampai 4600 mhz sementara DDR5 berada pada jangkauan 4800 sampai 8400mhz. Informasi *type* berdasarkan kecepatan ini dilakukan dengan *hardcode*.

Ketiga, implementasi *Rule-based Filtering* pada data dimana beberapa kriteria untuk membuat komponen yang dipilih menjadi relevan dengan mengaplikasikan aturan bisnis dengan *Rule-based Filtering*. Aturan ini diaplikasikan ke beberapa komponen inti.

1) Aturan untuk CPU

- a) Tidak ada CPU Server
- b) RCPU-1: If name contains {XEON, EPYC, OPTERON, THREADRIPPER, ATOM, CELERON, PENTIUM} ⇒ discard
- c) Harus socket terbaru
- d) RCPU-2: socket ∈ {AM5, AM4, LGA1700, LGA1200} ⇒ retain

2) Aturan untuk GPU

- a) Hanya ada GPU Desktop
RGPU-1: If chipset contains {NVS, QUADRO, FIREPRO, TESLA, TITAN, GRID} ⇒ discard
- b) Skor G3D lebih tinggi dari 15000 untuk GPU modern
RGPU-2: G3D_Mark > 15000 ⇒ retain

3. Aturan untuk RAM

- a) Kapasitas minimum untuk komputer modern
RRAM-1: capacity_gb ≥ 16 ⇒ retain
- b) DDR4 harus memiliki kecepatan minimal 3200mhz dengan harga kurang dari sama dengan 1200000
RRAM-2: (speed_mhz ≥ 3200) ∧ (capacity_gb/price_idr < 1,200,000) ⇒ retain
- c) DDR5 harus memiliki kecepatan minimal 5200mhz dengan harga kurang dari sama dengan 1500000
RRAM-3: (speed_mhz ≥ 5200) ∧ (capacity_gb/price_idr < 1,500,000) ⇒ retain

4. Aturan untuk *storage*

- a) *Storage* atau penyimpanan harus SSD
RST-1: type = SSD ⇒ retain

5. Aturan untuk PSU

- a) PSU harus bersertifikasi efisiensi penggunaan daya
RPSU-1: $\text{efficiency} \in \{\text{Bronze, Gold, Platinum, Titanium}\} \Rightarrow \text{retain}$
- b) Tidak ada PSU dari merek yang dituliskan karena kredibilitas yang tidak diketahui
RPSU-2: If name contains any of $\{\text{LOGISYS, DIABLOTEK, APEVIA, RAIDMAX, SOLID GEAR, KENTEK, COOLMAX, THERMAL MASTER, GIGAMAX, LEPA, GAMDIAS, SEGOTEP, ROSEWILL}\} \Rightarrow \text{discard}$

Aturan-aturan diatas merupakan aturan yang dapat diberlakukan untuk membuat komputer modern saat ini. Kriteria tersebut diaplikasikan pada dataset yang telah diproses sebelumnya untuk mendapatkan hasil dataset yang baru untuk dipilih menggunakan algoritma Greedy.

Keempat, sistem tidak membagi anggaran secara merata melainkan menggunakan alokasi anggaran secara dinamis. Sebagian budget dialokasikan untuk CPU dan GPU. Setelah itu menggunakan Rule-based filtering memilih PSU yang sesuai dengan kebutuhan daya kedua komponen inti tersebut. Anggaran untuk *case* dan *storage* dicadangkan di awal untuk memastikan bahwa *case* dan *storage* selalu terbeli sebesar 4% dan *storage* 5%. Pada kasus *case*, jika tidak terdapat *case* dengan harga tersebut maka akan menggunakan sisa budget global yang ada. Anggaran terakhir digunakan untuk membeli *motherboard* dan RAM

Kelima, Komponen yang dipilih oleh algoritma Greedy adalah komponen optimal lokal dengan memperhatikan kriteria atau aturan maksimum dan minimum pada suatu fitur pada komponen tersebut sebagai contoh CPU dan GPU dipilih berdasarkan skor tertinggi dan harga termurah. Pada setiap proses Greedy terdapat aturan atau Rules yang harus diberlakukan pada setiap komponen untuk menghasilkan komponen yang optimal.

```

Algorithm: Sort_Component_By_Rules
Input:
  candidates      - list of component candidates
  component_type  - type of component (e.g., cpu, gpu, ram, etc.)
  sort_rules      - mapping of component_type to sorting criteria

Output:
  sorted_candidates - ordered list of candidates

BEGIN
  rule = sort_rules[component_type]
  IF rule EXISTS THEN
    sorted_candidates = sort candidates
                        by rule.attributes
                        using rule.order
  ELSE
    sorted_candidates = sort candidates
                        by price_idx (ascending)
  END IF
  RETURN sorted_candidates
END

```

Gambar 2. Pseudocode Pemilihan Greedy.

Pemilihan komponen dimulai dari GPU atau *video card* karena budget terbesar dan performa komputer ditentukan dari sini. Dalam perakitan komputer modern maka GPU dengan chipset tipe RTX lebih diinginkan daripada jenis lain karena terdapat lebih banyak fitur seperti

DLSS dan fitur AI lainnya.. Maka dari itu, GPU dengan chipset RTX lebih diprioritaskan daripada jenis GPU lain. Jika tidak ada GPU dengan chipset RTX dengan harga yang sesuai maka sistem akan kembali memilih dari pool semua GPU. Argumen Greedy yang diperhatikan untuk komponen GPU adalah skor tertinggi dengan harga termurah.

```

Algorithm 3: GPU Selection Rule
BEGIN
  RTX.Pool ← all GPUs where chipset contains "RTX"
  GPU ← select_best(RTX.Pool, caps["gpu"], "gpu")

  IF GPU = None THEN
    GPU ← select_best(all GPUs, caps["gpu"], "gpu")
  END IF

  IF GPU ≠ None THEN
    partial_build("video-card") ← GPU
  ELSE
    warnings.append("Tidak dapat menemukan GPU yang sesuai dengan budget.")
  END IF
END
    
```

Gambar 3. Pseudocode Aturan Pemilihan GPU.

Tidak ada kriteria tertentu untuk pemilihan CPU karena tidak ada fitur unggulan yang unik dari berbagai merek sehingga argumen untuk pemilihan serakah atau Greedy adalah skor tertinggi dengan harga termurah.

Pada sistem ini, PSU yang dipilih adalah PSU yang mempunyai sertifikasi efisiensi daya mulai dari bronze sampai platinum. Cara pemilihan PSU dalam pemilihan Greedy atau serakah melalui beberapa kondisi berikut

- 1) Minimal watt dari PSU dihitung dengan $(TDP\ GPU + TDP\ CPU) * 1.4$. Perhitungan tersebut didapat berdasarkan referensi dari berbagai PSU Calculator seperti Corsair [12] dimana menambahkan kebutuhan daya GPU dan CPU lalu menambahkan perkiraan daya lebih untuk keamanan dan stabilitas kebutuhan daya untuk komputer.
- 2) Jika CPU merupakan seri K, skor GPU diatas atau sama dengan 30 ribu, dan budget diatas 25 juta maka minimal daya PSU adalah 1000 watt untuk mencegah pemilihan PSU yang tidak maksimal. Pada budget tinggi, semua komponen setidaknya harus mempunyai spesifikasi yang lebih tinggi walaupun kebutuhannya tidak sebanyak itu untuk efisiensi dan kestabilan yang lebih optimal.

Argumen Greedy yang digunakan adalah harga termurah.

```

Algorithm: Select_Power_Supply
BEGIN
IF "video-card" < partial_build AND "cpu" < PARTIAL_BUILD THEN
min_watts = (TDP_GPU + TDP_CPU) * 1.5
is_high_end_cpu = ("K" < CPU_name)
is_high_end_gpu = (GPU_performance_score > 30000)
IF is_high_end_cpu AND is_high_end_gpu AND budget > 25,000,000 THEN
min_watts = max(min_watts, 1000)
END IF
psu = select_best(
  components["power-supply"],
  budget * 0.15,
  component_type = "psu",
  constraints = { wattage > min_watts },
  order = ascending
)
IF psu = NULL THEN
partial_build["power-supply"] = psu
ELSE
warnings.append("Could not find a suitable PSU.")
END IF
END IF
END

```

Gambar 4. Pseudocode Aturan Pemilihan PSU.

Terdapat beberapa tingkatan atau seri *motherboard* untuk memadai penggunaan berat dengan dilengkapi VRM (*Voltage Regulator Module*) yang lebih tangguh dan kemampuan overclocking yang lebih ekstrim. Untuk orang biasa, fitur unggulan tersebut biasanya tidak akan dipakai namun untuk budget besar memakai *motherboard* seri tinggi dapat memaksimalkan penggunaan budget yang ada dan mendapatkan komputer yang memiliki performa lebih tinggi lagi.

- 1) Sebelum pemilihan *motherboard*, maka diperlukan daftar *motherboard* yang sesuai dengan pemilihan komponen sementara yang ada. Jika budget diatas atau sama dengan 25 juta dan CPU adalah seri K untuk Intel maka daftar *motherboard* yang dipilih adalah tipe 3 seri Z untuk Intel atau seri X untuk AMD.
- 2) Untuk budget diatas 15 juta, *motherboard* dengan DDR5 lebih diprioritaskan daripada DDR4.
- 3) Jika kondisi tersebut tidak terpenuhi maka akan memilih dari semua daftar *motherboard* yang ada.

Argumen yang sesuai untuk pemilihan Greedy adalah tingkat atau *tier* paling tinggi dengan harga termurah.

```

Algorithm: Select_Motherboard
BEGIN
  IF budget > 25,000,000 AND is_x_cpu AND components["mobo_highclass"] is not empty THEN
    mobo_pool = components["mobo_highclass"]
  ELSE IF budget > 17,000,000 AND components["mobo_mid_range"] is not empty THEN
    mobo_pool = components["mobo_mid_range"]
  END IF

  IF budget > 15,000,000 AND components["mobo_low"] is not empty THEN
    mobo_subset = filter(mobo_pool where is_ssr = TRUE)
    IF mobo_subset is not empty THEN
      mobo_pool = mobo_subset
    END IF
  END IF

  selected_motherboard = select_best(
    mobo_pool,
    support_budget,
    "ssr",
    constraint = { socket = build["cpu"].socket }
  )

  IF selected_motherboard is NULL THEN
    selected_motherboard = select_best(
      components["mobo_all"],
      support_budget,
      "ssr",
      constraint = { socket = build["cpu"].socket }
    )
  END IF

  RETURN selected_motherboard
END

```

Gambar 5. Pseudocode Aturan Pemilihan motherboard.

Pada sistem ini, pemilihan RAM bergantung pada besarnya budget. Jika budget di atas 35 juta maka maksimal RAM yang didapat adalah 64GB untuk memaksimalkan biaya ke komponen lainnya. Budget di atas atau sama dengan 15 juta maka mendapatkan setidaknya kapasitas 32GB. Argumen pemilihan Greedy untuk RAM adalah speed atau kecepatan ram tertinggi dengan harga terendah.

```

Algorithm: Select_Memory_Module
BEGIN
  IF support_budget > 35 AND "motherboard" & partial_build THEN
    ram_type = partial_build["motherboard"].ram_type
    selected_ram = NULL

    IF "ram_high_capacity" & components THEN
      selected_ram = select_best(
        components["ram_high_capacity"],
        support_budget,
        "ram",
        constraint = { type = ram_type }
      )
    END IF

    IF selected_ram IS NULL AND "ram_mid_capacity" & components THEN
      selected_ram = select_best(
        components["ram_mid_capacity"],
        support_budget,
        "ram",
        constraint = { type = ram_type }
      )
    END IF

    IF selected_ram IS NOT NULL THEN
      partial_build["memory"] = selected_ram
    END IF
  END IF
END

```

Gambar 6. Pseudocode Aturan Pemilihan RAM.

Komponen penyimpanan atau *Storage* adalah komponen yang subjektif sesuai kebutuhan. Seseorang mungkin butuh lebih banyak atau tidak membutuhkan banyak kapasitas karena bukan seorang pengguna berat. Namun dapat dipastikan bahwa kapasitas besar menjadi pilihan utama khususnya pada budget yang besar. Maka dari itu ada kondisi khusus untuk hal tersebut. Maksimal kapasitas adalah 4TB untuk budget di atas 30 juta, ini menjadi batas supaya sistem tidak memilih *storage* yang berlebih atau diluar pemakaian wajar. Jika budget di atas atau sama dengan 15 juta maka maksimal kapasitas adalah 2TB untuk menjaga budget

terdistribusikan dengan baik ke semua komponen. Selain itu maka maksimal kapasitas SSD atau *storage* adalah 1TB.

Case merupakan komponen opsional yang harganya bervariasi tergantung dari bahan, kompleksitas desain, dan ukuran. Untuk konteks sistem ini, sistem memilih case berdasarkan dari anggaran yang sudah diberikan di awal yaitu 4% atau jika masih terdapat banyak sisa budget global maka sistem akan menggunakan budget tersebut untuk pemilihan *case*.

Setelah seluruh komponen dipilih melalui proses seleksi sebelumnya, sistem melakukan tahap pengecekan kompatibilitas untuk memastikan bahwa setiap komponen yang direkomendasikan dapat saling bekerja dengan baik secara teknis. Pada tahap ini, sistem memverifikasi kesesuaian antara CPU dan motherboard berdasarkan jenis socket yang digunakan, mencocokkan tipe RAM dengan dukungan motherboard, serta memeriksa kesesuaian form factor motherboard dengan casing yang dipilih. Pengecekan kompatibilitas ini bertujuan untuk meminimalkan potensi kesalahan perakitan dan memastikan bahwa konfigurasi yang dihasilkan dapat digunakan tanpa kendala teknis. Setelah semua komponen dinyatakan kompatibel, sistem menyusun build PC akhir dengan menggabungkan seluruh komponen terpilih ke dalam satu paket rakitan lengkap.

Selanjutnya, sistem melakukan perhitungan total harga dengan menjumlahkan estimasi biaya dari masing-masing komponen untuk memastikan bahwa total biaya tetap berada dalam batas anggaran yang telah ditentukan oleh pengguna. Pada tahap akhir, sistem menyajikan hasil rekomendasi dalam bentuk daftar komponen lengkap beserta harga masing-masing, total estimasi biaya keseluruhan, serta catatan tambahan apabila terdapat kondisi khusus, seperti keterbatasan pilihan komponen atau potensi ketidaksesuaian terhadap alokasi anggaran. Penyajian hasil ini diharapkan dapat membantu pengguna dalam memahami konfigurasi rakitan PC yang direkomendasikan secara jelas dan komprehensif.

Pembahasan

Pengujian dilakukan dengan mencoba berbagai input budget. Budget minimal adalah 5 juta dan berhasil mendapatkan rekomendasi walaupun tidak dapat menemukan budget untuk membeli GPU. Pada kasus 100 juta, sistem dapat merekomendasikan komponen dengan seri tertinggi dan mampu menggunakan hampir semua budget dengan optimal karena masih menyisakan banyak budget.

Tabel 1. Sampel Pengujian.

Input	Hasil
Rp. 5.000.000	<p>CPU: AMD Athlon II X2 240 2 cores Boost: nan GHz Price: Rp542.432</p> <p>GPU : Tidak dapat menemukan GPU</p> <p>MOTHERBOARD: ASRock B550M-HDV Socket: AM4 Form: Micro ATX Price: Rp1.655.815</p> <p>RAM: Thermaltake TOUGHRAM XG RGB 16 GB 16GB DDR4 @ 4400MHz Price: Rp1.236.480</p> <p>STORAGE: Orico Y-20 Capacity: 128 GB Type: SSD Price: Rp202.215</p> <p>PSU: Tidak dapat menemukan PSU</p> <p>CASE: Silverstone Precision PS15 RGB Price: Rp1.345.960</p> <p>Total Harga: Rp4.982.902 Sisa : Rp 17.098</p>
Rp 18.000.000	<p>CPU: Intel Core i7-12700KF Price: Rp3.219.815 12 cores Boost: 5.0 GHz</p> <p>GPU: PNY Dual Fan OC Price: Rp6.504.031 Chipset: GeForce RTX 5060 Ti VRAM: 8.0GB</p> <p>MOTHERBOARD: ASRock Z790 PG SONIC Price: Rp2.943.816 Socket: LGA1700 Form: ATX</p> <p>RAM: Patriot Viper Xtreme 5 32 GB Price: Rp2.023.815 32GB DDR5 @ 7600MHz</p> <p>STORAGE: Silicon Power A55 Price: Rp846.215 Capacity: 1 TB Type: SSD</p> <p>PSU: SAMA GT Price: Rp1.471.816 850W gold</p> <p>CASE: Zalman CUBIX Rp699.016</p> <p>Total Harga: Rp17.708.524 Sisa : Rp 291.476</p>
Rp 15.000.000	<p>CPU: AMD Ryzen 7 5800XT Price: Rp2.576.000 8 cores Boost: 4.8 GHz</p> <p>GPU: Zotac GAMING SOLO Price: Rp5.519.816 Chipset: GeForce RTX 5060 VRAM: 8.0GB</p>

	<p>MOTHERBOARD: ASRock B550M-HDV Price: Rp1.655.815 Socket: AM4 Form: Micro ATX</p> <p>RAM: G.Skill Trident Z RGB 32 GB Price: Rp1.655.815 32GB DDR4 @ 4000MHz</p> <p>STORAGE: Orico Y-20 Price: Rp469.015 Capacity: 512 GB Type: SSD</p> <p>PSU: SAMA GT Price: Rp1.471.816 850W gold</p> <p>CASE: Aerocool Trinity Mini V2 Rp1.649.743</p> <p>Total Harga: Rp14.998.020 Sisa : Rp 1.980</p>
Rp 100.000.000	<p>CPU: AMD Ryzen 9 9950X3D 16 cores Boost: 5.7 GHz Price: Rp11.959.816</p> <p>GPU: Gigabyte WINDFORCE Chipset: GeForce RTX 5090 VRAM: 32.0GB Price: Rp36.799.816</p> <p>MOTHERBOARD: ASRock X870 Pro RS Socket: AM5 Form: ATX Price: Rp3.311.815</p> <p>RAM: V-Color TRA524G80S840R3Q 96 GB 96GB DDR5 @ 8000MHz Price: Rp25.612.615</p> <p>STORAGE: TEAMGROUP QX Price: Rp3.367.015 Capacity: 4 TB Type: SSD</p> <p>PSU: Montech CENTURY II Price: Rp2.022.159 1050W gold</p> <p>CASE: Phanteks NV7 Price: Rp3.974.031</p> <p>Total Harga: Rp. 87.047.267 Sisa : Rp. 12.952.733</p>

Pada pengujian 18 juta dan 15 juta, dapat disimpulkan bahwa sistem dapat mengaplikasikan algoritma serakah atau Greedy dengan baik karena dapat hampir menyentuh budget khususnya budget 15 juta yang dapat menyisakan sekitar 2 ribu. Untuk memvalidasi kombinasi rakitan komputer ini, kita dapat mengambil referensi dari suatu website terkenal untuk merakit komputer pada bagian rekomendasinya. Referensi diambil dari PCPartPicker pada bagian panduan untuk budget 24 juta 923 ribu

Tabel 2. Tabel Perbandingan dengan PCPartPicker.

Input	Hasil
Rp. 24.923.000	<p>CPU: Intel Core i5-14600KF Price: Rp3.642.831 14 cores Boost: 5.3 GHz</p> <p>GPU : MSI SHADOW 2X OC Price: Rp11.039.816 Chipset: GeForce RTX 5070 VRAM: 12.0GB</p> <p>MOTHERBOARD: ASRock Z790 PG SONIC Price: Rp2.943.816 Socket: LGA1700 Form: ATX</p> <p>RAM: Patriot Viper Xtreme 5 32 GB Price: Rp2.483.816 32GB DDR5 @ 8200MHz</p> <p>STORAGE : Inland Platinum Price: Rp993.415 Capacity: 1 TB Type: SSD</p> <p>PSU: SAMA GT Price: Rp1.471.816 850W gold</p> <p>CASE: Zalman S3 Price: Rp975.015</p> <p>Total Harga: Rp. 23.550.525 Sisa: Rp 1.372.475</p>

Berdasarkan dari kombinasi yang dihasilkan sistem dapat merekomendasikan kombinasi komponen yang dapat menyaingi panduan dari PCPartPicker. GPU yang dipilih oleh sistem sama dengan yang dimiliki oleh PCPartPicker. Dengan perbandingan ini, sistem dapat memilih kombinasi terbaik dari berbagai rules atau kriteria terkait pada tiap komponen. Perbedaan terbesar adalah pemilihan CPU, PCPartPicker menggunakan brand dari AMD sedangkan sistem menghasilkan CPU dari Intel seri i5. Sistem membandingkan dari berbagai kemungkinan CPU bahwa Intel Core i5-14600KF adalah pilihan terbaik dari segi performa yang dilihat dari CPU_Mark dan dari segi harga termurah.

Tabel 3. Kelebihan dan Kekurangan Sistem.

Aspek	Deskripsi
Kelebihan	<ol style="list-style-type: none"> Meminimalisir kesalahan kompatibilitas dengan menggunakan pendekatan Rule-Based Filtering untuk mencegah pemilihan komponen yang tidak kompatibel. Memastikan komponen rakitan PC memiliki performa yang baik sesuai anggaran yang ditetapkan menggunakan algoritma Greedy.
Kekurangan	<ol style="list-style-type: none"> Hasil rekomendasi masih terbatas pada jumlah dan variasi data di dalam dataset yang digunakan. Sistem belum memiliki fitur machine learning untuk beradaptasi dengan tren performa pasar yang berubah atau preferensi pengguna yang kompleks.

Hasil pengujian mengkonfirmasi bahwa kombinasi Rule-Based dan Greedy Algorithm adalah pendekatan yang efektif untuk masalah rakitan PC berbasis anggaran. Komponen Rule-Based memberikan jaminan validitas teknis, sementara Greedy Algorithm memberikan solusi optimal dalam kerangka ekonomi. Namun, untuk meningkatkan akurasi lebih lanjut, dataset perlu diperkaya agar sistem mempunyai referensi komponen yang lebih banyak. Pada penelitian ini, ruang penelitian dibatasi hanya menggunakan algoritma tersebut untuk mengetahui efektivitas kombinasi algoritma sederhana Rule-based filtering dan Greedy Algorithm, hasil yang didapat dari menggunakan *machine learning* dapat lebih baik lagi karena dapat menggunakan teknik unsupervised learning untuk melakukan rekomendasi (Tundjungsari, 2024). Sistem seharusnya bekerja lebih baik dalam lingkup *machine learning* karena terdapat spesifikasi untuk tiap komponen dan bisa membuat rekomendasi menggunakan algoritma Collaborative Filtering sesuai rakitan yang sudah dibuat oleh orang lain atau berdasarkan spesifikasi atau keperluan yang mirip (Tundjungsari, 2024).

Input sistem dibatasi minimal budget adalah 5 juta karena keterbatasan komponen serta harganya. Maksimal input adalah tidak terhingga namun menyisakan banyak budget tersisa. Pada kasus input karakter, sistem menolak dengan menyatakan error bahwa input tidak valid dan mengharuskan pengguna memasukkan angka.

5. KESIMPULAN DAN SARAN

Dari penelitian dan implementasi yang sudah dilakukan, sistem rekomendasi komponen rakit PC yang dikembangkan menggunakan algoritma Greedy dan Rule-based Filtering mampu memberikan rekomendasi komponen secara efisien, kompatibel, dan sesuai kebutuhan pengguna. Algoritma Greedy dapat dengan efektif menentukan kombinasi komponen terbaik berdasarkan berbagai aturan dan kriteria yang telah ditentukan atau hardcode sehingga menghasilkan hasil atau output kombinasi komponen PC yang sesuai dengan anggaran yang telah dimasukkan. Sementara itu, Rule-based Filtering berperan penting dalam sistem untuk melakukan filtering terhadap komponen yang tidak kompatibel antar komponen CPU, GPU, Case, Motherboard, serta kapasitas PSU. Rule-based Filtering berperan besar untuk menyaring dan membuat calon daftar komponen yang akan dipilih oleh algoritma Greedy.

Hasil pengujian menunjukkan bahwa sistem sederhana yang dibangun oleh algoritma Greedy dan Rule-based ini dapat memberikan rekomendasi dengan cukup baik karena dapat menyentuh hampir budget pengguna. Memaksimalkan budget yang diberikan dan dapat tidak melewati batas budget yang ada.

Adapun beberapa pengembangan yang dapat dilakukan untuk penelitian selanjutnya adalah penggunaan machine learning untuk meningkatkan kemampuan sistem memahami preferensi pengguna bukan dari anggaran saja namun dari berbagai aspek seperti benchmark suatu aplikasi atau suatu game untuk kebutuhan gaming. Berdasarkan algoritma Unsupervised Learning dapat digunakan untuk membuat rekomendasi, dalam konteks ini dapat digunakan untuk membuat rekomendasi komponen komputer yang sesuai berdasarkan spesifikasi yang ada pada komputer dan performanya. Penggunaan integrasi API dengan e-commerce atau situs harga terpercaya juga dapat dilakukan guna menghasilkan output yang lebih real-time dan ketersediaan stok supaya pengguna dapat langsung melihat produk yang direkomendasikan.

DAFTAR REFERENSI

- Ardhana, V. Y. P., & Mulyodiputro, M. D. (2023). Pelatihan perakitan komputer untuk meningkatkan keterampilan bagi santri di Ponpes Al Mutmainnah. *Jurnal Pengabdian Literasi Digital Indonesia*, 2(2), 49–54.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to Algorithms* (3rd ed.). MIT Press.
- Dima, J., Hamzah, M. S., Tallo, C. G., & Fallo, D. Y. A. (2025). Tinjauan literatur tentang pemanfaatan algoritma greedy untuk pencarian jalur terpendek. *Jurnal Kridatama Sains Dan Teknologi*, 7(1), 519–528.
- Ginting, J. (2022). Pelatihan perakitan komputer sebagai dasar pengenalan teknologi bagi calon pelaku UMKM. *Jurnal Abdimas HAWARI: Jurnal Pengabdian Kepada Masyarakat*, 2(1), 26–31.
- Maulana, V. A. (2025). Pengembangan sistem rekomendasi untuk simulasi rakit komputer menggunakan algoritma genetika berbasis website (Skripsi, Universitas Pembangunan Nasional Veteran Jawa Timur).
- Ndruru, R., & Lubis, Y. F. A. (2025). Implementasi algoritma greedy untuk rekomendasi objek wisata di Pulau Nias berbasis website. *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, 4(2), 1145–1152.
- Pokutta, S., Singh, M., & Torrico, A. (2020). On the unreasonable effectiveness of the greedy algorithm: Greedy adapts to sharpness. In *Proceedings of the International Conference on Machine Learning* (pp. 7772–7782).
- Preeth, E. N., Mulerickal, F. J. P., Paul, B., & Sastri, Y. (2015). Evaluation of Docker containers based on hardware utilization. In *2015 International Conference on Control Communication & Computing India (ICCC)* (pp. 697–700).
- Ricci, F., Rokach, L., & Shapira, B. (2015). *Recommender Systems Handbook* (2nd ed.). Springer. <https://doi.org/10.1007/978-1-4899-7637-6>

- Roihan, A., Nasution, K., & Siambaton, M. Z. (2022). Implementasi algoritma greedy kombinasi dengan perulangan pada aplikasi penjadwalan praktikum. *Sudo Jurnal Teknik Informatika*, 1(2), 42–50.
- Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- Saputra, R. D. (2024). Sistem rekomendasi pemilihan paket rakitan komputer menggunakan metode knowledge base di CV Innovision (Skripsi, Universitas Duta Bangsa Surakarta).
- Siregar, R. N., Aulia, R., & Siregar, R. (2025). Implementasi algoritma greedy pada rute wisata Danau Tao Desa Batang Onang Baru berbasis website. *Explorer*, 5(1), 33–41.
- Tundjungsari, V. (2024). *Dasar machine learning* (Edisi revisi). Deepublish.
- Utomo, M. T., & Mustafidah, H. (2016). Penentuan spesifikasi komputer berdasarkan kebutuhan pemakai dan harga menggunakan basis data fuzzy. *JUITA: Jurnal Informatika*, 4(1), 28–36.