Teori Himpunan untuk Rekayasa Perangkat Lunak Modul Belajar dengan Visual yang Dapat Dibaca Tanpa Melihat

Muhammad Riduwan Khafidi
Program Studi Teknik Rekayasa Perangkat Lunak
Politeknik Negeri Batam
4342611034@students.polibatam.ac.id

November 5, 2026

Abstrak

Modul ini mengajarkan teori himpunan sebagai alat kerja untuk memodelkan data dalam rekayasa perangkat lunak. Setiap konsep dibuka dengan analogi sehari-hari, dijelaskan mengapa ia ada, lalu diperagakan lewat contoh berangka yang ditulis langkah demi langkah. Sebelas visual dirancang hitam-putih dan disertai narasi yang menggambarkan bentuk, posisi, label, dan makna setiap bagian sehingga pembaca yang tidak dapat melihat gambar tetap dapat memahaminya. Modul ini juga memuat sebelas tabel ringkas, satu studi kasus delapan langkah, serta bagian Cara Berpikir dan Kalau Salah di setiap konsep kunci. Setelah membaca modul ini, pembaca dapat berpindah dari data mentah menuju keputusan yang dapat dihitung, diperiksa, dan dipertanggungjawabkan.

1 Pendahuluan

1.1 Analogi Pembuka: Gudang Arsip

Bayangkan sebuah gudang arsip. Di dalam gudang itu ada banyak lemari. Setiap lemari punya label, misalnya Lemari Mahasiswa, Lemari Mata Kuliah, dan Lemari Nilai. Setiap lemari berisi laci-laci. Setiap laci diberi nomor. Setiap dokumen hanya boleh tinggal di satu laci yang benar.

Jika dokumen yang sama dimasukkan ke tiga laci sekaligus, tiga hal buruk terjadi. Pertama, pencarian menjadi lambat karena petugas harus memeriksa tiga laci. Kedua, laporan menjadi salah karena satu dokumen dihitung tiga kali. Ketiga, ketika dokumen itu diperbarui, dua salinan lain tertinggal dalam keadaan lama.

Teori himpunan adalah aturan gudang arsip itu. Ia menentukan apa yang termasuk, apa yang tidak termasuk, dan bagaimana kumpulan yang berbeda digabungkan atau dipisahkan tanpa membuat salinan yang salah.

1.2 Mengapa Topik Ini Penting

Sistem perangkat lunak menyimpan data sebagai kumpulan objek. Daftar pengguna, daftar produk, daftar transaksi, dan daftar hak akses semuanya adalah kumpulan. Tanpa teori himpunan, pengembang sulit menjawab pertanyaan seperti berapa pengguna unik hari ini, siapa yang memakai dua platform sekaligus, dan mengapa total pelanggan tidak boleh dihitung dengan menjumlahkan semua kategori secara langsung. Teori himpunan memberi bahasa yang presisi untuk pertanyaan-pertanyaan itu.

1.3 Cara Membaca Modul Ini

Setiap bagian mengikuti pola yang sama. Pertama, analogi konkret. Kedua, alasan mengapa konsep itu diperlukan. Ketiga, definisi formal. Keempat, contoh berangka. Kelima, visual dengan narasi pengganti gambar. Keenam, tabel ringkas. Bagian terakhir menutup dengan tiga prinsip dan daftar rujukan.

Jika pembaca hanya punya waktu singkat, bacalah setiap bagian "Mengapa Ini Penting" dan tabel ringkas di akhir setiap bagian. Jika pembaca ingin memahami secara mendalam, bacalah berurutan.

1.4 Peta Pembahasan

Bagian kedua membahas apa itu himpunan dan mengapa larik biasa tidak cukup. Bagian ketiga membahas cara menyajikan himpunan. Bagian keempat membahas kardinalitas. Bagian kelima membahas himpunan khusus dan relasi antar himpunan. Bagian keenam membahas operasi dasar. Bagian ketujuh membahas prinsip inklusi-eksklusi. Bagian kedelapan membahas himpunan fuzzy. Bagian kesembilan menyandingkan seluruh konsep. Bagian kesepuluh memakai konsep pada studi kasus dashboard pengguna. Bagian kesebelas dan keduabelas memberi daftar kesalahan umum dan lembar rujukan cepat. Bagian ketigabelas menutup dengan tiga prinsip.

2 Apa Itu Himpunan dan Mengapa Larik Tidak Cukup

2.1 Analogi: Daftar Tamu versus Daftar Kursi

Bayangkan sebuah pesta. Ada dua daftar yang bisa dibuat panitia. Daftar pertama adalah daftar kursi: kursi nomor satu diduduki siapa, kursi nomor dua diduduki siapa, dan seterusnya. Daftar kedua adalah daftar tamu: siapa saja orang yang hadir, tanpa memedulikan kursi mana yang ia duduki.

Jika Ani berpindah kursi tiga kali, daftar kursi mencatat Ani tiga kali. Daftar tamu mencatat Ani satu kali. Untuk menghitung berapa orang yang hadir, daftar tamu yang benar. Daftar kursi menjawab pertanyaan yang berbeda, yaitu kursi mana yang kosong.

Larik mirip daftar kursi. Himpunan mirip daftar tamu. Larik peduli pada posisi. Himpunan peduli pada keunikan.

2.2 Definisi Formal

Himpunan adalah kumpulan objek berbeda yang terdefinisi dengan jelas. Objek di dalam himpunan disebut elemen. Notasi himpunan memakai kurung kurawal. Jika U adalah himpunan pengguna unik, maka $U = \{\text{USR01}, \text{USR02}, \text{USR03}\}$ berarti U berisi tiga pengguna berbeda. Penulisan $x \in U$ berarti x adalah elemen U. Penulisan $x \notin U$ berarti x bukan elemen U [**?**].

Ada tiga kata kunci dalam definisi itu. Pertama, berbeda. Himpunan tidak menyimpan duplikat. Kedua, terdefinisi dengan jelas. Harus ada aturan yang menentukan apakah suatu objek termasuk atau tidak. Ketiga, objek. Elemen himpunan bisa berupa angka, nama, baris tabel, atau apa pun yang bisa dibedakan.

2.3 Mengapa Larik Tidak Cukup

Larik memuat duplikat. Larik $[\text{USR01}, \text{USR02}, \text{USR01}]$ memiliki tiga posisi, tetapi hanya dua pengguna unik, yaitu USR01 dan USR02. Jika larik itu dipakai untuk menghitung pengguna aktif, hasilnya tiga, padahal seharusnya dua. Kesalahan ini terlihat sepele, tetapi dalam sistem analitik ia berarti laporan pengguna aktif terlalu besar sepertiga.

Himpunan menghapus duplikat secara otomatis. Himpunan dari larik di atas adalah $\{\text{USR01}, \text{USR02}\}$. Kardinalitasnya dua. Inilah dasar semua perhitungan pengguna unik dalam sistem analitik [**?**]. Perhatikan Gambar **??**.

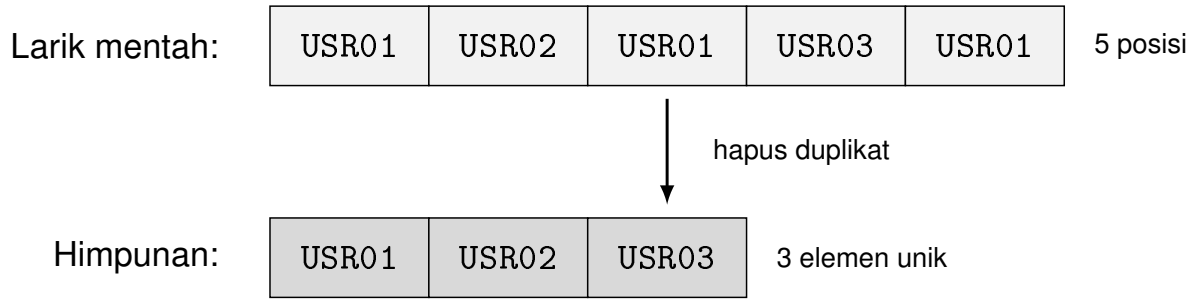


Figure 1. Larik mentah dengan duplikat dan himpunan unik yang dihasilkannya.

Cara membaca Gambar **??**: gambar ini terdiri dari dua baris. Baris atas adalah deretan lima kotak kecil yang mewakili lima posisi larik. Dari kiri ke kanan, isi kotak adalah USR01, USR02, USR01, USR03, USR01. Di kanan deretan tertulis "lima posisi". Baris bawah adalah deretan tiga kotak kecil yang mewakili tiga elemen himpunan. Dari kiri ke kanan, isi kotak adalah USR01, USR02, USR03. Di kanan deretan tertulis "tiga elemen unik". Di antara dua baris itu ada panah pendek yang mengarah ke bawah dengan label "hapus duplikat". Inti pesan gambar ini adalah satu larik dengan lima posisi hanya memiliki tiga pengguna unik, dan himpunanlah yang menyimpan angka tiga itu, bukan larik.

2.4 Cara Berpikir: Kapan Data Disebut Himpunan

Saat menghadapi data mentah, ajukan tiga pertanyaan berurutan. Pertama, apakah posisi penting? Jika posisi penting, misalnya nomor urut antrean, maka data itu larik, bukan himpunan. Kedua, apakah duplikat boleh ada? Jika duplikat tidak boleh ada, maka data itu himpunan. Ketiga, apakah ada aturan yang menentukan anggota? Jika ada aturan, misalnya "semua pengguna yang login hari ini", maka data itu himpunan dengan notasi pembentuk.

Jika ketiga pertanyaan itu dijawab, representasi data sudah pasti. Pengembang tidak perlu menebak. Ia hanya perlu menjawab tiga pertanyaan itu.

2.5 Mengapa Ini Penting untuk Rekayasa Perangkat Lunak

Basis data relasional memakai kunci utama untuk menjamin tidak ada duplikat. Kunci utama adalah janji bahwa setiap baris di tabel berbeda dari baris lain. Janji itu adalah janji himpunan. Tanpa konsep himpunan, pengembang sulit menjelaskan mengapa tabel tanpa kunci utama cepat menjadi kacau.

Sistem analitik memakai operator DISTINCT untuk menghapus duplikat sebelum menghitung. Operator itu juga janji himpunan. Tanpa konsep himpunan, pengembang sulit menjelaskan mengapa DISTINCT memperlambat query pada data besar.

Pengujian perangkat lunak memakai kombinasi unik dari masukan. Setiap kombinasi adalah elemen himpunan. Tanpa konsep himpunan, penguji sulit menjelaskan mengapa dua kasus uji yang identik tidak menambah cakupan.

2.6 Ringkasan

Tabel **??** menyandingkan larik dan himpunan pada aspek yang paling sering membingungkan.

Table 1. Perbandingan larik dan himpunan.

Aspek	Larik	Himpunan
Duplikat	boleh ada	tidak boleh ada
Urutan	penting	tidak penting
Pertanyaan	posisi mana	anggota mana
Kardinalitas	panjang larik	banyak elemen unik
Analogi	daftar kursi	daftar tamu
Contoh kode	`list`	`set`

3 Cara Menyajikan Himpunan

3.1 Analogi: Empat Cara Menulis Resep

Bayangkan seorang koki menulis resep. Ia bisa menulis daftar bahan secara lengkap, misalnya "telur, tepung, gula, garam". Ia bisa menulis aturan, misalnya "semua bahan yang ada di rak bumbu". Ia bisa memakai nama baku, misalnya "bumbu dasar putih". Ia juga bisa menggambar tata letak meja saji.

Keempat cara itu menyampaikan hal yang sama dengan tingkat ketelitian yang berbeda. Cara pertama paling jelas untuk himpunan kecil. Cara kedua paling ringkas untuk himpunan besar. Cara ketiga paling cepat untuk himpunan yang sudah punya nama. Cara keempat paling mudah dipahami saat hubungan antar himpunan perlu ditunjukkan.

3.2 Empat Cara Menyajikan Himpunan

Cara pertama adalah enumerasi. Semua elemen ditulis di antara kurung kurawal. Contoh: $A = \{101, 102, 105, 108, 112\}$. Cara ini cocok jika elemen sedikit dan sudah diketahui.

Cara kedua adalah simbol baku. Sejumlah himpunan sudah punya simbol tetap. $\mathbb{N} = \{1, 2, 3, \ldots\}$ adalah himpunan bilangan asli. $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ adalah himpunan bilangan bulat. $\mathbb{Q}$ adalah himpunan bilangan rasional. $\mathbb{R}$ adalah himpunan bilangan riil. $\mathbb{C}$ adalah himpunan bilangan kompleks. Cara ini cocok untuk himpunan yang sudah dikenal luas.

Cara ketiga adalah notasi pembentuk himpunan. Elemen ditentukan oleh aturan. Contoh: $B = \{x \mid 100 \leq x \leq 115 \text{ dan } x \text{ habis dibagi } 3\}$. Tanda $\mid$ dibaca "dengan syarat". Bagian kiri tanda adalah elemen. Bagian kanan adalah syarat. Tanda koma di dalam syarat dibaca "dan". Cara ini cocok untuk himpunan besar dengan aturan yang jelas.

Cara keempat adalah diagram Venn. Himpunan digambar sebagai daerah. Cara ini cocok untuk menunjukkan hubungan antar himpunan. Perhatikan Gambar **??**.

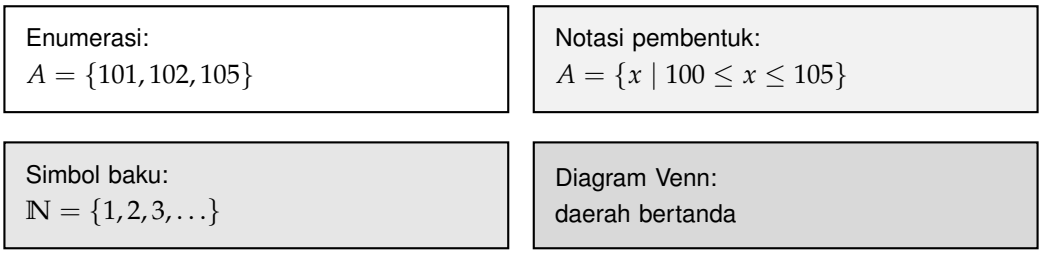


Figure 2. Empat cara menyajikan himpunan.

Cara membaca Gambar **??**: gambar ini terdiri dari empat kotak persegi panjang yang tersusun dua baris dan dua kolom. Kotak kiri atas berisi tulisan "Enumerasi" diikuti contoh $A = \{101, 102, 105\}$. Kotak kanan atas berisi tulisan "Notasi pembentuk" diikuti contoh $A = \{x \mid 100 \leq x \leq 105\}$. Kotak kiri bawah berisi tulisan "Simbol baku" diikuti contoh $\mathbb{N} = \{1, 2, 3, \ldots\}$.

Kotak kanan bawah berisi tulisan "Diagram Venn" diikuti keterangan "daerah bertanda". Latar keempat kotak berbeda tingkat abu-abu, dari paling terang di kiri atas sampai paling gelap di kanan bawah. Gradasi itu hanya membedakan posisi, bukan menyatakan makna. Inti pesan gambar ini adalah himpunan yang sama bisa ditulis dalam empat cara, dan pemilihan cara bergantung pada kebutuhan.

3.3 Mengapa Ini Penting

Dalam basis data, enumerasi setara dengan tabel berisi baris eksplisit. Notasi pembentuk setara dengan query SELECT WHERE. Simbol baku setara dengan view yang sudah dinamai. Diagram Venn setara dengan diagram relasi antar tabel. Pengembang yang mengenali empat padanan ini dapat berpindah antara tabel, query, dan diagram tanpa bingung.

3.4 Ringkasan

Tabel **??** merangkum kapan setiap cara dipakai.

Table 2. Kapan memakai setiap cara penyajian himpunan.

Cara	Kapan dipakai	Padanan di basis data
Enumerasi	himpunan kecil	tabel eksplisit
Simbol baku	himpunan yang sudah dikenal	view bernama
Notasi pembentuk	himpunan besar dengan aturan	SELECT WHERE
Diagram Venn	hubungan antar himpunan	diagram relasi tabel

4 Kardinalitas: Menghitung Tanpa Duplikat

4.1 Analogi: Panitia Menghitung Tamu

Bayangkan panitia sebuah acara. Panitia berdiri di pintu masuk dengan sebuah papan dan sebuah pulpen. Setiap tamu yang masuk, panitia menanyakan namanya. Jika nama itu belum ada di papan, panitia menuliskannya. Jika nama itu sudah ada, panitia tidak menulis lagi, karena tamu itu sudah terhitung.

Di akhir acara, panitia tidak perlu menghitung berapa kali setiap orang keluar-masuk toilet. Panitia hanya perlu menghitung berapa nama berbeda yang tertulis di papan. Itulah kardinalitas.

4.2 Definisi Formal

Kardinalitas adalah banyaknya elemen berbeda dalam himpunan berhingga. Notasinya $|A|$ atau $n(A)$. Jika log mentah berisi $[\text{USR01}, \text{USR05}, \text{USR02}, \text{USR01}, \text{USR03}, \text{USR05}, \text{USR01}, \text{USR04}, \text{USR02}]$, maka himpunan

pengguna uniknya adalah $\{\text{USR01}, \text{USR05}, \text{USR02}, \text{USR03}, \text{USR04}\}$. Kardinalitas himpunan itu adalah lima, bukan sembilan, karena empat kemunculan berulang tidak dihitung lagi [**?**].

4.3 Contoh Berangka yang Ditulis Langkah Demi Langkah

Misalkan sebuah sistem menerima log selama satu jam. Log mentah berisi sembilan baris. Kita hitung kardinalitasnya dengan cara panitia di atas.

Langkah pertama, siapkan papan kosong. Papan saat ini berisi himpunan kosong.

Langkah kedua, baca baris pertama, USR01. Papan belum memuat USR01. Tulis USR01. Papan sekarang berisi $\{\text{USR01}\}$.

Langkah ketiga, baca baris kedua, USR05. Papan belum memuat USR05. Tulis USR05. Papan sekarang berisi $\{\text{USR01}, \text{USR05}\}$.

Langkah keempat, baca baris ketiga, USR02. Papan belum memuat USR02. Tulis USR02. Papan sekarang berisi $\{\text{USR01}, \text{USR05}, \text{USR02}\}$.

Langkah kelima, baca baris keempat, USR01. Papan sudah memuat USR01. Tidak tulis apa-apa. Papan tetap berisi tiga elemen.

Lanjutkan langkah ini sampai baris kesembilan. Setelah baris ketujuh dan kesembilan (USR01 dan USR02) tidak ditulis lagi, papan akhir berisi lima elemen. Kardinalitasnya lima.

Rasio penghematan adalah $5/9 \approx 0{,}56$, artinya hanya sekitar 56 persen dari log mentah yang benar-benar informasi baru. Jika sistem menerima satu juta baris log per jam, dan rasio penghematan sama, maka pengguna unik per jam hanya sekitar 560 ribu. Angka ini penting untuk kapasitas server.

4.4 Kalau Salah: Apa yang Terjadi Jika Kardinalitas Diabaikan

Misalkan sebuah dashboard menampilkan total kunjungan hari ini sebesar 300 ribu. Jika dashboard itu menjumlahkan panjang larik log, angka 300 ribu itu adalah jumlah kunjungan, bukan jumlah pengunjung. Jika rasio pengunjung unik terhadap kunjungan adalah 1 banding 3, maka pengunjung unik sebenarnya hanya 100 ribu.

Manajer yang membaca dashboard akan mengira ada 300 ribu orang yang tertarik pada produk. Ia akan memesan kapasitas server untuk 300 ribu orang. Padahal yang dibutuhkan hanya kapasitas untuk 100 ribu orang. Kesalahan ini mahal, dan satu-satunya cara mencegahnya adalah menghitung kardinalitas himpunan, bukan panjang larik.

4.5 Mengapa Ini Penting untuk Rekayasa Perangkat Lunak

Sistem analitik menghitung pengguna unik dengan operator DISTINCT. Tanpa DISTINCT, pengguna yang membuka halaman tiga kali akan terhitung tiga kali. Dengan DISTINCT, ia terhitung satu kali. Kardinalitas adalah ukuran resmi dari apa yang dihitung DISTINCT.

Basis data memakai kardinalitas untuk memilih strategi query. Kolom dengan kardinalitas tinggi, misalnya email, cocok diberi indeks. Kolom dengan kardinalitas rendah, misalnya jenis kelamin, kurang cocok diberi indeks karena indeks tidak banyak membantu. Keputusan ini didasarkan pada kardinalitas, bukan pada ukuran kolom.

4.6 Ringkasan

Tabel **??** merangkum istilah dan contoh.

Table 3. Ringkasan kardinalitas.

Istilah	Makna	Contoh
Larik mentah	posisi berurutan, boleh duplikat	9 baris log
Himpunan unik	elemen berbeda tanpa duplikat	5 pengguna
Kardinalitas	banyak elemen unik	$\|U\| = 5$
Operator SQL	DISTINCT	`SELECT DISTINCT user`

5 Himpunan Khusus dan Relasi Antar Himpunan

5.1 Analogi: Hak Akses di Sistem Akademik

Bayangkan sebuah sistem akademik dengan tiga peran: Admin, Dosen, dan Mahasiswa. Admin bisa input, edit, hapus, dan lihat. Dosen bisa input dan lihat. Mahasiswa hanya bisa lihat. Hak akses setiap peran adalah himpunan. Hak akses Mahasiswa adalah himpunan bagian dari hak akses Dosen. Hak akses Dosen adalah himpunan bagian dari hak akses Admin. Inilah relasi subset.

5.2 Himpunan Kosong dan Subset

Himpunan kosong adalah himpunan tanpa elemen. Notasinya $\varnothing$ atau $\{\}$. Himpunan A disebut himpunan bagian dari B jika setiap elemen A juga elemen B. Notasinya $A \subseteq B$. Contoh: $\{1, 2, 3\} \subseteq \{1, 2, 3, 4, 5\}$. Jika Admin punya hak akses $\{\text{input}, \text{edit}, \text{hapus}, \text{lihat}\}$ dan Mahasiswa punya hak akses $\{\text{lihat}\}$, maka himpunan hak akses Mahasiswa adalah subset dari himpunan hak akses Admin [**?**]. Perhatikan Gambar **??**.

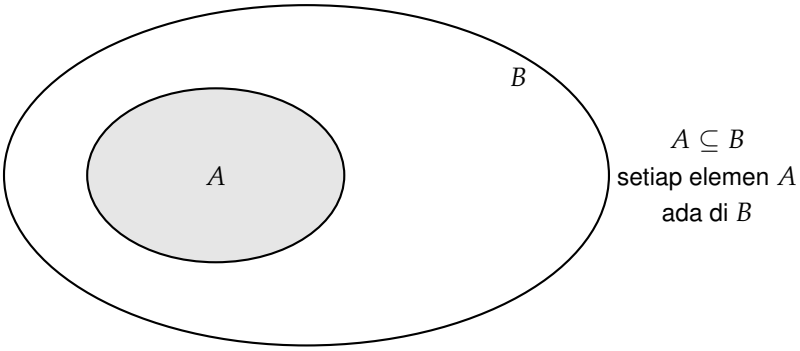


Figure 3. Subset A di dalam superset B.

Cara membaca Gambar **??**: gambar ini menampilkan dua elips. Elips besar berada di tengah dan diberi label B di sisi kanan atas. Di dalam elips besar itu ada elips kecil yang diberi label A. Elips kecil seluruhnya berada di dalam elips besar, tidak ada bagian yang keluar. Di sebelah kanan gambar, di luar kedua elips, tertulis $A \subseteq B$ dengan keterangan "setiap elemen A ada di B". Elips kecil diberi latar abu-abu muda sedangkan elips besar berlatar putih, supaya batas keduanya terlihat jelas. Inti pesan gambar ini adalah satu himpunan yang seluruhnya berada di dalam himpunan lain disebut subset.

5.3 Himpunan Kuasa

Himpunan kuasa dari A adalah himpunan yang semua elemennya adalah himpunan bagian dari A, termasuk himpunan kosong dan A sendiri. Notasinya $P(A)$ atau 2^A. Jika $A = \{a, b\}$, maka $P(A) = \{\varnothing, \{a\}, \{b\}, \{a, b\}\}$. Jika $|A| = n$, maka $|P(A)| = 2^n$. Konsep ini dipakai dalam pengujian perangkat lunak untuk menghasilkan seluruh kombinasi konfigurasi yang mungkin [**?**]. Perhatikan Gambar **??**.

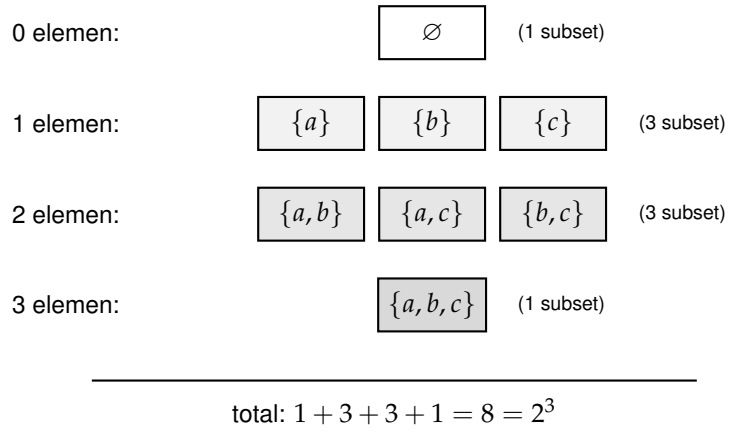


Figure 4. Himpunan kuasa dari $\{a, b, c\}$ berisi delapan subset, tersusun per jumlah elemen.

Cara membaca Gambar **??**: gambar ini menampilkan empat baris kotak yang disusun berdasarkan jumlah elemen di dalam subset. Baris paling atas berlabel "0 elemen" dan berisi satu kotak, yaitu $\varnothing$. Di kanannya tertulis "(1 subset)". Baris kedua berlabel "1 elemen" dan berisi tiga kotak, yaitu $\{a\}$, $\{b\}$, dan $\{c\}$. Di kanannya tertulis "(3 subset)". Baris ketiga

berlabel "2 elemen" dan berisi tiga kotak, yaitu $\{a, b\}$, $\{a, c\}$, dan $\{b, c\}$. Di kanannya tertulis "(3 subset)". Baris keempat berlabel "3 elemen" dan berisi satu kotak, yaitu $\{a, b, c\}$. Di kanannya tertulis "(1 subset)". Di bawah keempat baris ada garis pemisah dan tulisan "total: 1+3+3+1=8=2^3". *Intipesangambariniadalahhimpunandengantigaelemenmemilikidelapansubset, dandelapanitum*

5.4 Himpunan Sama dan Ekivalen

Himpunan A sama dengan B jika dan hanya jika $A \subseteq B$ dan $B \subseteq A$. Himpunan A ekivalen dengan B jika kardinalitas keduanya sama, walaupun elemennya berbeda. Contoh sama: $A = \{0, 1\}$ dan $B = \{x \mid x(x-1) = 0\}$ karena keduanya berisi 0 dan 1. Contoh ekivalen: $A = \{1, 3, 5, 7\}$ dan $B = \{a, b, c, d\}$ karena $|A| = |B| = 4$ **[?]**. Ringkasan relasi antar himpunan diberikan pada Tabel **??**.

Table 4. Ringkasan relasi antar himpunan.

Relasi	Syarat	Contoh
Subset	setiap elemen A ada di B	$\{1, 2\} \subseteq \{1, 2, 3\}$
Superset	B memuat semua elemen A	$\{1, 2, 3\} \supseteq \{1, 2\}$
Sama	$A \subseteq B$ dan $B \subseteq A$	$\{0, 1\} = \{x \mid x(x-1) = 0\}$
Ekivalen	$\|A\| = \|B\|$	$\{1, 3, 5, 7\}$ dan $\{a, b, c, d\}$
Himpunan kuasa	semua subset dari A	$P(\{a, b\}) = \{\varnothing, \{a\}, \{b\}, \{a, b\}\}$

5.5 Mengapa Ini Penting

Dalam sistem hak akses berbasis peran, relasi subset menjamin hierarki izin. Jika Mahasiswa bukan subset dari Dosen, maka ada hak akses yang dimiliki Mahasiswa tetapi tidak dimiliki Dosen. Kondisi itu biasanya bug. Dengan konsep subset, pengembang dapat menulis pengujian otomatis yang memeriksa relasi itu.

Dalam pengujian perangkat lunak, himpunan kuasa memberi daftar lengkap kombinasi konfigurasi. Jika ada tiga saklar fitur, maka ada delapan kombinasi hidup-mati. Himpunan kuasa memastikan tidak ada kombinasi yang terlewat.

6 Operasi Dasar pada Himpunan

6.1 Analogi: Filter di Aplikasi Pesan

Bayangkan sebuah aplikasi pesan dengan dua label: Label Kerja dan Label Keluarga. Setiap pesan boleh memiliki satu label, dua label, atau tidak sama sekali. Aplikasi menyediakan empat tombol filter. Tombol pertama menampilkan pesan yang berlabel Kerja dan Keluarga sekaligus. Tombol kedua menampilkan pesan yang berlabel Kerja atau Keluarga. Tombol ketiga menampilkan pesan yang tidak berlabel Kerja. Tombol keempat menampilkan pesan berlabel Kerja yang tidak berlabel Keluarga.

Keempat tombol itu adalah empat operasi dasar pada himpunan: irisan, gabungan, komplemen, dan selisih.

6.2 Cara Berpikir: Memilih Operasi yang Tepat

Sebelum memilih operasi, tanyakan lima pertanyaan ini secara berurutan. Pertama, apakah saya ingin elemen yang ada di kedua himpunan? Jika ya, pakai irisan. Kedua, apakah saya ingin elemen yang ada di salah satu himpunan, tanpa memedulikan apakah di kedua atau tidak? Jika ya, pakai gabungan. Ketiga, apakah saya ingin elemen yang tidak ada di satu himpunan tertentu? Jika ya, pakai komplemen. Keempat, apakah saya ingin elemen yang ada di himpunan pertama tetapi tidak di himpunan kedua? Jika ya, pakai selisih. Kelima, apakah saya ingin elemen yang ada di salah satu himpunan tetapi tidak di keduanya? Jika ya, pakai beda simetris.

Urutan pertanyaan ini penting. Banyak kesalahan terjadi karena orang langsung memakai irisan padahal yang dimaksud adalah gabungan, atau sebaliknya. Dengan lima pertanyaan ini, pemilihan operasi menjadi tidak ambigu.

6.3 Irisan, Gabungan, dan Komplemen

Irisan $A \cap B$ berisi elemen yang ada di A dan di B. Gabungan $A \cup B$ berisi elemen yang ada di A atau di B. Komplemen $\overline{A}$ berisi elemen semesta U yang bukan elemen A.

Contoh irisan: $A = \{2, 4, 6, 8, 10, 11\}$ dan $B = \{4, 10, 11, 14, 18\}$, maka $A \cap B = \{4, 10, 11\}$ karena hanya tiga angka itu yang muncul di kedua himpunan. Contoh gabungan: $A = \{2, 5, 8\}$ dan $B = \{7, 5, 22\}$, maka $A \cup B = \{2, 5, 7, 8, 22\}$ karena semua angka dari kedua himpunan digabung, dan angka 5 hanya ditulis sekali. Contoh komplemen: $U = \{1, 2, \ldots, 9\}$ dan $A = \{1, 3, 5, 7, 9\}$, maka $\overline{A} = \{2, 4, 6, 8\}$ karena empat angka genap itulah yang ada di semesta tetapi tidak di A [**?**]. Perhatikan Gambar **??**.

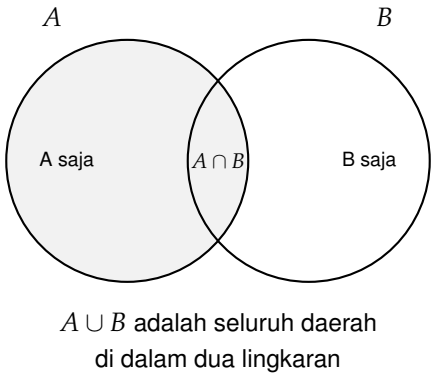


Figure 5. Irisan dan gabungan dua himpunan.

Cara membaca Gambar **??**: gambar ini menampilkan dua lingkaran yang saling berpotongan. Lingkaran pertama berada di kiri, label A diletakkan di atas sisi kiri lingkaran, di luar area lingkaran. Lingkaran kedua berada di kanan, label B diletakkan di atas sisi kanan lingkaran,

juga di luar area lingkaran. Kedua lingkaran saling menimpa sekitar sepertiga lebar masing-masing, sehingga terbentuk tiga daerah. Daerah kiri hanya tertutup lingkaran A, dan di dalamnya tertulis "A saja". Daerah tengah tertutup kedua lingkaran, dan di dalamnya tertulis $A \cap B$. Daerah kanan hanya tertutup lingkaran B, dan di dalamnya tertulis "B saja". Di bawah gambar, di luar lingkaran, tertulis "$A \cup B$ adalah seluruh daerah di dalam dua lingkaran". Lingkaran kiri diberi latar abu-abu sangat muda, sedangkan lingkaran kanan berlatar putih. Inti pesan gambar ini adalah irisan adalah daerah tengah, sedangkan gabungan adalah seluruh daerah di dalam dua lingkaran.

6.4 Selisih dan Beda Simetris

Selisih $A - B$ berisi elemen A yang bukan elemen B. Beda simetris $A \oplus B$ berisi elemen yang ada di A atau di B, tetapi tidak di keduanya. Dalam basis data relasional, selisih setara dengan perintah EXCEPT atau NOT IN. Jika $A = \{\text{User1}, \text{User2}, \text{User3}, \text{User5}\}$ adalah pengguna aktif dan $B = \{\text{User2}, \text{User5}\}$ adalah pengguna yang sudah membayar, maka $A - B = \{\text{User1}, \text{User3}\}$ adalah pengguna aktif yang belum membayar [**?**]. Perhatikan Gambar **??**.



Figure 6. Selisih dan beda simetris dua himpunan.

Cara membaca Gambar **??**: gambar ini terdiri dari dua subgambar yang diletakkan berdampingan. Subgambar kiri berisi dua lingkaran berpotongan. Lingkaran kiri diberi latar abu-abu dan diberi label $A - B$ di dalamnya, sedangkan lingkaran kanan berlatar putih dan diberi label B di dalamnya. Di bawah subgambar tertulis "selisih". Subgambar kanan juga berisi dua lingkaran berpotongan. Lingkaran kiri dan kanan keduanya diberi latar abu-abu, dan di daerah tengah yang berpotongan ada label $A \oplus B$. Di bawah subgambar tertulis "beda simetris". Inti pesan gambar ini adalah selisih hanya menyisakan satu sisi, sedangkan beda simetris menyisakan kedua sisi tanpa daerah tengah.

6.5 Produk Kartesian

Produk kartesian $A \times B$ adalah himpunan semua pasangan berurutan (a, b) dengan $a \in A$ dan $b \in B$. Jika $A = \{\text{Chrome}, \text{Firefox}\}$ dan $B = \{\text{Windows}, \text{Linux}, \text{MacOS}\}$, maka $A \times B$ berisi enam pasangan. Dalam pengujian perangkat lunak, produk kartesian dipakai untuk menghasilkan kombinasi browser dan sistem operasi yang harus diuji [**?**]. Perhatikan Gambar **??**.

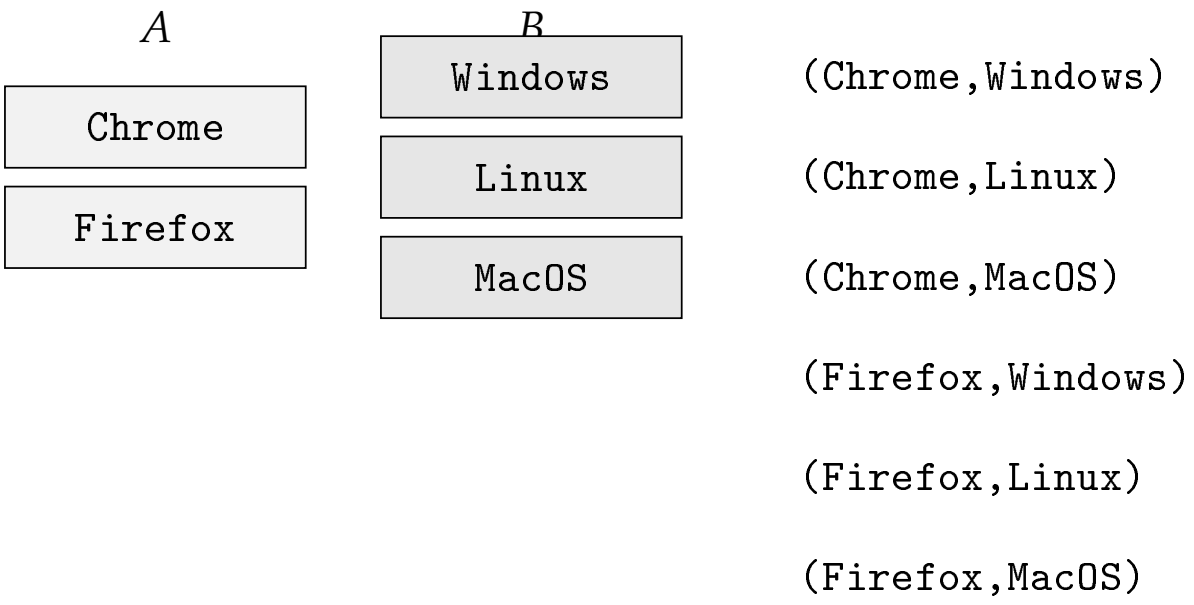


Figure 7. Produk kartesian browser dan sistem operasi.

Cara membaca Gambar **??**: gambar ini terdiri dari tiga kolom. Kolom paling kiri berjudul *A* dan berisi dua kotak kecil: kotak atas bertuliskan Chrome, kotak bawah bertuliskan Firefox. Kolom tengah berjudul *B* dan berisi tiga kotak kecil: kotak atas bertuliskan Windows, kotak tengah bertuliskan Linux, kotak bawah bertuliskan MacOS. Kolom paling kanan berisi enam tulisan pasangan, yaitu (Chrome,Windows), (Chrome,Linux), (Chrome,MacOS), (Firefox,Windows), (Firefox,Linux), dan (Firefox,MacOS). Setiap pasangan adalah kombinasi satu browser dengan satu sistem operasi. Inti pesan gambar ini adalah produk kartesian menghasilkan seluruh kombinasi yang mungkin antara dua himpunan.

6.6 Mengapa Ini Penting

Dalam basis data, irisan dipakai untuk JOIN, gabungan untuk UNION, komplemen untuk NOT IN, dan selisih untuk EXCEPT. Produk kartesian dipakai untuk CROSS JOIN. Pengembang yang mengenali padanan ini dapat membaca query SQL dengan mata seorang ahli himpunan.

Dalam pengujian perangkat lunak, produk kartesian menghasilkan matriks uji lengkap. Jika ada dua browser dan tiga sistem operasi, maka ada enam kasus uji. Jika angka itu dilipatkan dengan lima versi database, maka ada tiga puluh kasus uji. Perkalian ini berasal dari produk kartesian.

6.7 Ringkasan

Tabel **??** merangkum semua operasi.

Table 5. Ringkasan operasi dasar pada himpunan.

Operasi	Notasi	Makna	Padanan SQL
Irisan	$A \cap B$	elemen yang ada di A dan B	INNER JOIN
Gabungan	$A \cup B$	elemen yang ada di A atau B	UNION
Komplemen	$\overline{A}$	elemen semesta yang bukan A	NOT IN
Selisih	$A - B$	elemen A yang bukan B	EXCEPT
Beda simetris	$A \oplus B$	elemen di salah satu, tidak keduanya	XOR
Produk kartesian	$A \times B$	semua pasangan berurutan	CROSS JOIN

7 Prinsip Inklusi-Eksklusi

7.1 Analogi: Menghitung Siswa yang Ikut Ekstrakurikuler

Bayangkan sebuah sekolah dengan tiga ekstrakurikuler: Pramuka, Basket, dan Musik. Kepala sekolah ingin tahu berapa siswa yang ikut setidaknya satu ekstrakurikuler. Jika kepala sekolah hanya menjumlahkan peserta Pramuka, peserta Basket, dan peserta Musik, hasilnya akan terlalu besar, karena siswa yang ikut dua ekstrakurikuler terhitung dua kali, dan siswa yang ikut tiga ekstrakurikuler terhitung tiga kali.

Prinsip inklusi-eksklusi adalah cara memperbaiki hitungan itu. Mula-mula jumlahkan semua peserta. Lalu kurangi peserta yang ikut dua ekstrakurikuler. Lalu tambahkan peserta yang ikut tiga ekstrakurikuler, karena mereka telah dikurangi terlalu banyak.

7.2 Dua Himpunan

Untuk dua himpunan berhingga, berlaku $|A \cup B| = |A| + |B| - |A \cap B|$. Contoh: A adalah pengguna Mobile App dengan $|A| = 3500$, B adalah pengguna Web dengan $|B| = 2100$, dan $|A \cap B| = 800$. Maka $|A \cup B| = 3500 + 2100 - 800 = 4800$ pengguna unik.

Angka 4800 adalah jawaban yang benar. Jika irisan tidak dikurangi, hasilnya 5600, yaitu terlalu besar sebesar 800. Kesalahan sebesar 800 pada satu hari akan menumpuk menjadi kesalahan besar pada laporan bulanan [**?**].

7.3 Kalau Salah: Apa yang Terjadi Jika Lupa Koreksi Irisan

Misalkan dashboard menampilkan tiga angka: total pengguna Mobile 3500, total pengguna Web 2100, dan total pengguna unik 5600. Angka 5600 itu diperoleh dari penjumlahan langsung tanpa koreksi. Namun angka 5600 tidak mungkin benar, karena total pengguna unik tidak boleh melebihi total pengguna yang tercatat di sistem. Jika sistem mencatat 5000 pengguna, maka dashboard melaporkan angka yang lebih besar dari jumlah orang yang ada di sistem. Pengguna dashboard akan langsung tahu ada yang salah, tetapi ia tidak akan tahu di mana kesalahannya.

Koreksi irisan mengubah 5600 menjadi 4800. Angka 4800 masuk akal karena lebih kecil dari 5000. Perbedaan 800 adalah pengguna yang memakai kedua platform, dan mereka hanya boleh dihitung satu kali. Inilah gunanya inklusi-eksklusi: ia membuat angka kembali masuk akal.

7.4 Tiga Himpunan

Untuk tiga himpunan, berlaku rumus $|N \cup D \cup V| = |N| + |D| + |V| - |N \cap D| - |N \cap V| - |D \cap V| + |N \cap D \cap V|$. Dengan data $|N| = 120$, $|D| = 80$, $|V| = 60$, $|N \cap D| = 40$, $|N \cap V| = 25$, $|D \cap V| = 15$, dan $|N \cap D \cap V| = 10$, hasilnya adalah $120 + 80 + 60 - 40 - 25 - 15 + 10 = 190$. Perhatikan Gambar **??**.

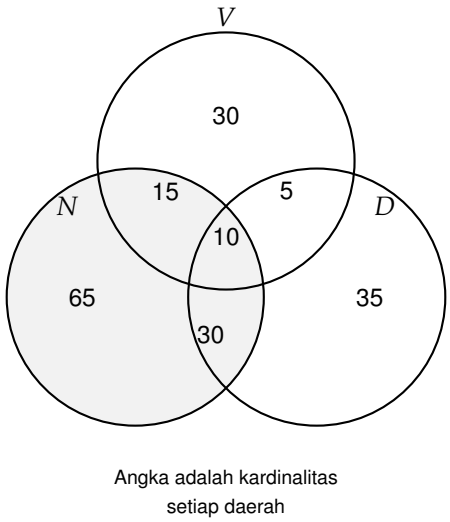


Figure 8. Diagram inklusi-eksklusi tiga himpunan.

Cara membaca Gambar **??**: gambar ini menampilkan tiga lingkaran berpotongan yang membentuk pola seperti tiga koin yang saling bertumpuk. Lingkaran kiri berlabel N, lingkaran kanan berlabel D, dan lingkaran atas berlabel V. Setiap daerah di dalam lingkaran diberi angka yang menyatakan kardinalitasnya. Daerah khusus N berisi 65, daerah khusus D berisi 35, dan daerah khusus V berisi 30. Daerah $N \cap D$ saja berisi 30, terletak di bagian bawah dekat pertemuan kedua lingkaran bawah. Daerah $N \cap V$ saja berisi 15, terletak di antara N dan V. Daerah $D \cap V$ saja berisi 5, terletak di antara D dan V. Daerah irisan ketiganya berisi 10, terletak di tengah, di antara ketiga lingkaran. Di bawah gambar tertulis "angka adalah kardinalitas setiap daerah". Jika seluruh angka dijumlahkan, hasilnya $65 + 30 + 35 + 30 + 15 + 5 + 10 = 190$, yaitu kardinalitas gabungan tiga himpunan. Inti pesan gambar ini adalah setiap angka di dalam diagram sudah menghitung daerah yang diwakilinya sendiri, bukan menghitung himpunan secara keseluruhan.

7.5 Mengapa Ini Penting

Setiap sistem analitik yang menjumlahkan beberapa kategori tanpa koreksi irisan akan melaporkan angka yang terlalu besar. Kesalahan ini tidak terlihat pada satu query, tetapi terlihat

pada dashboard yang menampilkan total pelanggan, pelanggan Mobile, pelanggan Web, dan pelanggan keduanya secara berdampingan. Jika total pelanggan tidak sama dengan gabungan, pengguna dashboard akan bingung.

7.6 Ringkasan

Tabel **??** merangkum rumus dan kesalahan umum.

Table 6. Ringkasan prinsip inklusi-eksklusi.

Kasus	Rumus dan makna
Dua himpunan	$\|A \cup B\| = \|A\| + \|B\| - \|A \cap B\|$
Tiga himpunan	$\|A \cup B \cup C\| = \|A\| + \|B\| + \|C\| - \|A \cap B\| - \|A \cap C\| - \|B \cap C\| + \|A \cap B \cap C\|$
Tujuan	menghindari penghitungan ganda pada daerah irisan
Kesalahan umum	menjumlahkan semua kategori tanpa mengurangi irisan

8 Himpunan Fuzzy

8.1 Analogi: Suhu Air di Pancuran

Bayangkan sebuah pancuran. Jika air terlalu dingin, kita menyebutnya dingin. Jika air terlalu panas, kita menyebutnya panas. Tetapi air pada suhu 40 derajat Celsius bisa disebut hangat, dan pada 45 derajat bisa disebut panas sedang. Batas dingin, hangat, dan panas tidak bertepi tajam. Itulah gagasan himpunan fuzzy: keanggotaan bisa sebagian, bukan hanya ya atau tidak.

8.2 Cara Berpikir: Kapan Memakai Fuzzy

Ajukan tiga pertanyaan. Pertama, apakah data aslinya biner, misalnya laki-laki atau perempuan, hidup atau mati? Jika biner, pakai crisp. Kedua, apakah data aslinya berupa derajat, misalnya tingkat kepuasan satu sampai sepuluh, tingkat kecurigaan spam nol sampai satu? Jika berupa derajat, pakai fuzzy. Ketiga, apakah keputusan akhirnya tetap biner, misalnya terima atau tolak? Jika ya, fuzzy masih berguna karena ia memberi urutan sebelum keputusan itu diambil. Urutan itulah yang hilang jika kita memakai crisp sejak awal.

8.3 Crisp dan Fuzzy

Himpunan crisp memakai keanggotaan biner. Nilai $\mu_A(x)$ hanya 0 atau 1. Himpunan fuzzy memakai derajat keanggotaan pada selang $[0, 1]$. Nilai 0,7 berarti elemen cenderung kuat menjadi anggota, tetapi tidak mutlak. Himpunan fuzzy diperkenalkan oleh Zadeh pada tahun 1965 untuk menangani ketidakpastian dan ekspresi linguistik seperti sedikit, lumayan, dan sangat [**?**]. Perhatikan Gambar **??**.

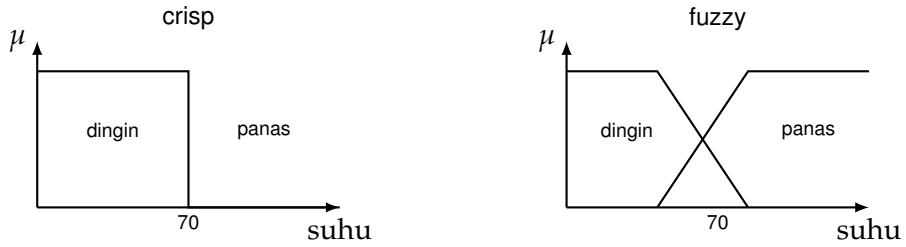


Figure 9. Perbandingan fungsi keanggotaan crisp dan fuzzy.

Cara membaca Gambar **??**: gambar ini terdiri dari dua grafik yang diletakkan berdampingan. Grafik kiri berlabel "crisp". Sumbu horizontal bertuliskan "suhu" dan sumbu vertikal bertuliskan "" (derajat keanggotaan). Kurva di grafik kiri berbentuk tangga: dari suhu nol sampai 70, garis mendatar di ketinggian 1. Pada suhu 70, garis turun mendadak ke 0. Dari suhu 70 sampai 100, garis mendatar di ketinggian 0. Di bawah bagian tinggi ada tulisan "dingin", dan di bawah bagian rendah ada tulisan "panas". Grafik kanan berlabel "fuzzy". Kurvanya berbentuk dua garis miring yang bersilangan di dekat suhu 70. Garis turun dari ketinggian 1 pada suhu rendah ke ketinggian 0 pada suhu tinggi, dan garis naik dari ketinggian 0 pada suhu rendah ke ketinggian 1 pada suhu tinggi. Kedua garis bertemu di dekat suhu 70 pada ketinggian setengah. Inti pesan gambar ini adalah crisp memakai batas tajam, sedangkan fuzzy memakai transisi bertahap.

8.4 Fungsi Keanggotaan Fuzzy

Fungsi keanggotaan fuzzy memetakan elemen ke nilai pada $[0, 1]$. Jika himpunan usia memiliki kategori MUDA, PAROBAYA, dan TUA, maka usia 40 dapat memiliki $\mu_{\text{MUDA}}(40) = 0{,}25$ dan $\mu_{\text{PAROBAYA}}(40) = 0{,}50$. Artinya, usia 40 lebih condong ke PAROBAYA daripada MUDA, tetapi masih memiliki derajat MUDA. Perhatikan Gambar **??**.

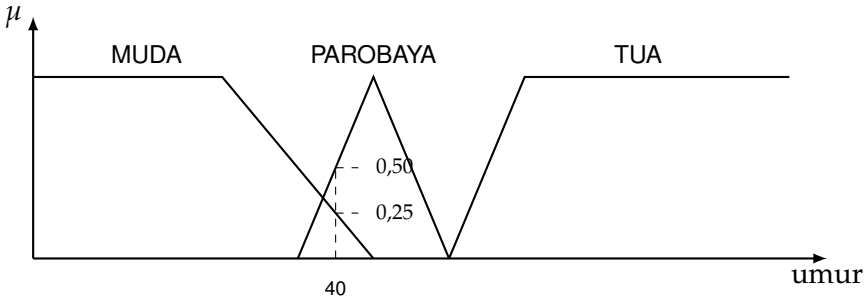


Figure 10. Fungsi keanggotaan fuzzy untuk kategori usia.

Cara membaca Gambar **??**: gambar ini menampilkan satu grafik dengan sumbu horizontal berlabel "umur" dan sumbu vertikal berlabel "". Di dalam grafik ada tiga bentuk yang mewakili tiga kategori. Bentuk pertama adalah trapesium yang menutupi rentang umur rendah, berlabel MUDA di atasnya. Bentuk kedua adalah segitiga yang menutupi rentang umur tengah, berlabel PAROBAYA di atasnya. Bentuk ketiga adalah trapesium yang menutupi rentang umur tinggi, berlabel TUA di atasnya. Ketiga bentuk saling tumpang tindih di tepinya. Pada umur 40, ada

satu garis putus-putus vertikal yang naik dari sumbu horizontal. Garis itu bercabang mendatar di ketinggian 0,6 dan di ketinggian 1,2. Cabang bawah diberi label 0,25 yang menyatakan nilai keanggotaan MUDA. Cabang atas diberi label 0,50 yang menyatakan nilai keanggotaan PAROBAYA. Di bawah sumbu horizontal ada label 40 yang menyatakan umur. Inti pesan gambar ini adalah pada umur 40, seseorang memiliki derajat keanggotaan di dua kategori sekaligus, dengan derajat yang berbeda.

8.5 Contoh Peringkat Rekomendasi

Jika Film A memiliki derajat suka 0,7, Film B memiliki derajat netral 0,5, dan Film C memiliki derajat suka sangat 1, maka urutan rekomendasi dari paling disukai adalah Film C, Film A, lalu Film B. Himpunan fuzzy lebih tepat daripada himpunan crisp karena preferensi pengguna tidak selalu ya atau tidak. Nilai 0,7 dan 0,5 memberi urutan yang lebih kaya daripada hanya suka dan tidak suka [**?**].

8.6 Kalau Salah: Apa yang Hilang Jika Memakai Crisp

Misalkan seorang pengguna memberi penilaian pada lima film: 1,0 untuk Film A, 0,9 untuk Film B, 0,5 untuk Film C, 0,4 untuk Film D, dan 0,3 untuk Film E.

Jika sistem memakai crisp dengan ambang 0,6, maka sistem hanya menyimpan dua kategori: Film A dan Film B adalah suka, Film C sampai Film E adalah tidak suka. Sistem kehilangan informasi bahwa pengguna lebih menyukai Film C daripada Film D, dan Film D daripada Film E. Ketika sistem harus merekomendasikan satu film pengganti jika Film A sudah ditonton, ia akan memilih Film B karena Film B satu-satunya yang tersisa di kategori suka. Padahal pengguna mungkin lebih cocok dengan Film C yang hanya sedikit di bawah ambang.

Jika sistem memakai fuzzy, urutan lengkap Film A sampai Film E tetap tersimpan. Sistem dapat merekomendasikan Film B sebagai pilihan utama dan Film C sebagai pilihan kedua. Pengguna menerima rekomendasi yang lebih masuk akal. Inilah nilai fuzzy: ia mempertahankan urutan yang hilang di crisp.

8.7 Mengapa Ini Penting

Sistem rekomendasi yang memakai crisp hanya bisa membedakan dua tingkat: suka atau tidak suka. Semua film yang disukai akan berada di peringkat yang sama. Sistem rekomendasi yang memakai fuzzy dapat membedakan sepuluh tingkat, seratus tingkat, atau bahkan seribu tingkat. Semakin halus derajatnya, semakin baik urutan yang dapat dihasilkan.

Sistem deteksi spam juga memakai fuzzy. Email tidak hanya ditandai spam atau bukan spam. Email diberi skor kecurigaan, misalnya 0,72, lalu dimasukkan ke folder karantina. Ambang batas skor ditentukan oleh kebijakan, bukan oleh kategori biner.

8.8 Ringkasan

Tabel **??** merangkum perbedaan crisp dan fuzzy.

Table 7. Perbandingan himpunan crisp dan fuzzy.

Aspek	Crisp	Fuzzy
Nilai keanggotaan	0 atau 1	bilangan real pada $[0, 1]$
Batas kategori	tegas	bertahap
Contoh suhu	dingin atau panas	sebagian dingin dan sebagian panas
Contoh rekomendasi	suka atau tidak suka	derajat suka 0,7
Kegunaan	aturan biner	penilaian bertingkat

9 Sintesis Seluruh Konsep

Tabel **??** menyandingkan semua konsep dalam satu pandangan. Kolom pertama menyebut lapisan konsep. Kolom kedua menyebut pertanyaan yang dijawab. Kolom ketiga menyebut alat matematis. Kolom keempat menyebut contoh penerapan.

Table 8. Sintesis konsep himpunan untuk rekayasa perangkat lunak.

Lapisan	Pertanyaan	Alat	Contoh penerapan
Dasar	apa saja anggota kumpulan ini	notasi himpunan dan kardinalitas	daftar pengguna unik
Relasi	apakah satu kumpulan bagian dari kumpulan lain	subset dan superset	hierarki hak akses
Operasi	bagaimana menggabungkan atau memisahkan kumpulan	irisan, gabungan, selisih	filter pengguna aktif
Hitung	berapa total unik tanpa penghitungan ganda	inklusi-eksklusi	pengguna Mobile dan Web
Ketidakpastian	seberapa kuat elemen masuk kategori	himpunan fuzzy	skor rekomendasi film

Sintesis dari Tabel **??** adalah satu pola berpikir. Data mentah diubah menjadi himpunan unik, lalu himpunan itu dioperasikan, lalu dihitung dengan koreksi irisan, dan jika perlu diberi derajat keanggotaan fuzzy. Pola ini berlaku dari analitik pengguna sampai sistem rekomendasi.

10 Studi Kasus: Dashboard Pengguna E-commerce

10.1 Langkah 1: Kumpulkan Data Mentah

Data yang tersedia adalah 10.000 pengguna, 6.000 pengguna Mobile App, 4.500 pengguna Web, 2.000 pengguna keduanya, 5.500 pengguna melakukan pembelian, dan 1.500 pengguna memakai Mobile App dan Web serta melakukan pembelian. Data mentah ini belum menjadi informasi karena batas antar kategori belum dinyatakan sebagai himpunan.

10.2 Langkah 2: Bentuk Himpunan Pengguna Unik

Bentuk M sebagai himpunan pengguna Mobile App, W sebagai himpunan pengguna Web, dan P sebagai himpunan pengguna yang melakukan pembelian. Kardinalitasnya adalah $|M| = 6000$, $|W| = 4500$, $|P| = 5500$, dan $|M \cap W| = 2000$. Himpunan menghapus duplikat sehingga satu pengguna yang membuka aplikasi tiga kali tetap dihitung satu.

10.3 Langkah 3: Definisikan Semesta Pembicaraan

Semesta U adalah seluruh pengguna yang tercatat pada periode analisis. Tanpa semesta, komplemen tidak dapat dihitung. Jika U adalah 10.000 pengguna, maka pengguna yang tidak memakai Mobile App dan tidak memakai Web adalah komplemen dari $M \cup W$. Namun nilai itu baru sah jika U benar-benar memuat semua pengguna yang relevan.

10.4 Langkah 4: Hitung Irisan

Irisan $M \cap W$ berisi pengguna yang memakai Mobile App dan Web. Nilainya diketahui 2.000. Angka ini menjadi koreksi utama karena pengguna di daerah itu akan dihitung dua kali jika $|M|$ dan $|W|$ dijumlahkan langsung.

10.5 Langkah 5: Hitung Gabungan

Gabungan $M \cup W$ berisi pengguna yang memakai salah satu atau kedua platform. Dengan inklusi-eksklusi, $|M \cup W| = 6000 + 4500 - 2000 = 8500$. Jadi ada 8.500 pengguna unik yang memakai Mobile App atau Web.

10.6 Langkah 6: Hitung Daerah Khusus

Pengguna hanya Mobile App adalah $|M - W| = |M| - |M \cap W| = 6000 - 2000 = 4000$. Pengguna hanya Web adalah $|W - M| = |W| - |M \cap W| = 4500 - 2000 = 2500$. Kedua angka ini menjawab pertanyaan dashboard tentang komposisi pengguna per platform.

10.7 Langkah 7: Hitung Beda Simetris

Pengguna yang memakai Mobile App atau Web tetapi tidak keduanya adalah $|M \oplus W| = |M \cup W| - |M \cap W| = 8500 - 2000 = 6500$. Angka ini juga sama dengan $4000 + 2500 = 6500$. Beda simetris berguna untuk kampanye yang menyasar pengguna yang baru memakai satu platform.

10.8 Langkah 8: Periksa Kelengkapan Data Tiga Himpunan

Untuk menghitung $|M \cup W \cup P|$, diperlukan $|M \cap P|$ dan $|W \cap P|$. Data yang tersedia hanya $|M \cap W \cap P| = 1500$. Jadi dashboard belum boleh menampilkan gabungan tiga himpunan secara final. Langkah yang benar adalah menandai data yang kurang, bukan mengarang angka. Inilah sikap akademik yang diuji oleh kasus ini [**?**]. Alur langkah studi kasus ditunjukkan pada Gambar **??**.

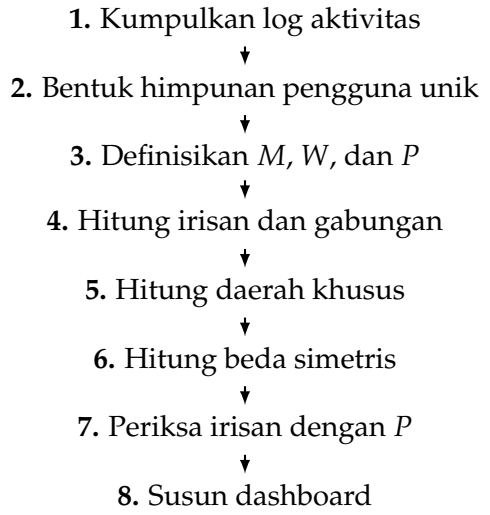


Figure 11. Alur delapan langkah studi kasus dashboard pengguna.

Cara membaca Gambar **??**: gambar ini menampilkan delapan kotak persegi panjang yang tersusun dari atas ke bawah. Setiap kotak berisi satu langkah, diawali angka tebal. Kotak pertama berisi "Kumpulkan log aktivitas". Kotak kedua berisi "Bentuk himpunan pengguna unik". Kotak ketiga berisi "Definisikan M, W, dan P". Kotak keempat berisi "Hitung irisan dan gabungan". Kotak kelima berisi "Hitung daerah khusus". Kotak keenam berisi "Hitung beda simetris". Kotak ketujuh berisi "Periksa irisan dengan P". Kotak kedelapan berisi "Susun dashboard". Setiap kotak dihubungkan ke kotak berikutnya oleh panah pendek yang mengarah ke bawah. Kotak-kotak itu memakai latar abu-abu sangat muda sehingga batasnya terlihat. Inti pesan gambar ini adalah proses penyusunan dashboard berjalan berurutan dari pengumpulan data sampai penyajian.

Table 9. Ringkasan hasil studi kasus dashboard pengguna.

Langkah	Operasi	Hasil
1	data mentah	10.000 pengguna
2	bentuk M, W, P	$\|M\| = 6000$, $\|W\| = 4500$, $\|P\| = 5500$
3	irisan $M \cap W$	2.000
4	gabungan $M \cup W$	8.500
5	hanya Mobile	4.000
6	hanya Web	2.500
7	beda simetris $M \oplus W$	6.500
8	gabungan tiga himpunan	belum final karena data kurang

11 Kesalahan Umum dan Cara Menghindarinya

Tabel **??** memuat kesalahan yang paling sering muncul saat memakai teori himpunan dalam analisis data. Setiap kesalahan disertai akibat dan cara memperbaikinya.

Table 10. Kesalahan umum pada penerapan teori himpunan.

Kesalahan	Akibat	Perbaikan
menghitung larik mentah sebagai himpunan	duplikat ikut terhitung	bentuk himpunan unik lebih dahulu
menjumlahkan $\|A\| + \|B\|$ tanpa irisan	hasil terlalu besar	kurangi $\|A \cap B\|$
menganggap subset sama dengan elemen	relasi salah	bedakan $x \in A$ dan $A \subseteq B$
lupa mendefinisikan semesta	komplemen tidak sah	tulis U secara eksplisit
mencampur notasi crisp dan fuzzy	makna nilai kabur	tetapkan domain nilai keanggotaan
menghitung $\|P(A)\| = n$	hasil salah	gunakan 2^n
mengabaikan data irisan tiga himpunan	gabungan tidak final	tandai data yang kurang
memakai warna untuk membedakan daerah	tidak terbaca hitam-putih	pakai label dan gradasi abu-abu
menulis $A - B$ sebagai $B - A$	arah selisih salah	periksa elemen mana yang dikurangi

Dua kesalahan yang paling sering adalah menjumlahkan kategori tanpa koreksi irisan dan melupakan semesta pembicaraan. Kesalahan pertama membuat dashboard terlalu optimistis. Kesalahan kedua membuat komplemen menjadi tidak bermakna. Keduanya dapat dicegah dengan menulis definisi himpunan sebelum menghitung.

12 Lembar Rujukan Cepat

Tabel **??** adalah lembar rujukan untuk ujian atau kerja. Baris pertama sampai baris terakhir mencakup seluruh notasi yang dipakai di modul ini.

Table 11. Lembar rujukan cepat teori himpunan.

Kebutuhan	Notasi atau langkah
menyatakan elemen	$x \in A$
menyatakan bukan elemen	$x \notin A$
menulis himpunan	$A = \{a, b, c\}$
menulis aturan	$A = \{x \mid \text{syarat}\}$
kardinalitas	$\lvert A \rvert$
himpunan kosong	$\varnothing$
subset	$A \subseteq B$
superset	$B \supseteq A$
himpunan kuasa	$P(A)$ dan $\lvert P(A) \rvert = 2^{\lvert A \rvert}$
irisan	$A \cap B$
gabungan	$A \cup B$
komplemen	$\overline{A}$
selisih	$A - B$
beda simetris	$A \oplus B$
produk kartesian	$A \times B$
inklusi-eksklusi dua himpunan	$\lvert A \cup B \rvert = \lvert A \rvert + \lvert B \rvert - \lvert A \cap B \rvert$
inklusi-eksklusi tiga himpunan	$\lvert A \cup B \cup C \rvert = \lvert A \rvert + \lvert B \rvert + \lvert C \rvert - \lvert A \cap B \rvert - \lvert A \cap C \rvert - \lvert B \cap C \rvert + \lvert A \cap B \cap C \rvert$
keanggotaan fuzzy	$\mu_A(x) \in [0, 1]$

13 Kesimpulan

Modul ini menunjukkan bahwa teori himpunan bukan bahan hafalan, melainkan alat kerja. Tiga prinsip penutup adalah sebagai berikut.

1. Nyatakan batas himpunan sebelum menghitung. Tanpa definisi yang jelas, angka apa pun bisa tampak benar.
2. Koreksi irisan sebelum menjumlahkan. Prinsip inklusi-eksklusi mencegah penghitungan ganda.
3. Gunakan derajat keanggotaan ketika data tidak biner. Himpunan fuzzy memberi urutan yang lebih jujur daripada kategori ya atau tidak.

Dengan tiga prinsip itu, pembaca dapat berpindah dari data mentah menuju dashboard yang dapat dipertanggungjawabkan, serta siap mempelajari topik berikutnya dalam seri matematika diskrit.

Referensi

References

[1] Rosen, K. H. (2019). *Discrete Mathematics and Its Applications* (8th ed.). McGraw-Hill Education.

[2] Munir, R. (2012). *Matematika Diskrit*. Informatika.

[3] Epp, S. S. (2010). *Discrete Mathematics with Applications* (4th ed.). Brooks Cole.

[4] Lipschutz, S., & Lipson, M. (2007). *Discrete Mathematics*. McGraw-Hill.

[5] Zadeh, L. A. (1965). Fuzzy Sets. *Information and Control*, 8(3), 338–353.

[6] Silberschatz, A., Korth, H. F., & Sudarshan, S. (2019). *Database System Concepts* (7th ed.). McGraw-Hill Education.

[7] Kleppmann, M. (2017). *Designing Data-Intensive Applications*. O'Reilly Media.

[8] Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson.