# **BAB II LANDASAN TEORI**

1. **Data Konsumen**

Data konsumen adalah informasi yang di terima saat konsumen berinteraksi dengan media bisnis. Media ini dapat berupa situs web, aplikasi seluler, halaman media sosial, halaman survei, kampanye, dan versi online atau offline lainnya dari upaya pemasaran bisnis [3].

Data konsumen penting karena memungkinkan bisnis untuk mengoptimalkan platform melalui pemahaman berbagai aspek interaksi konsumen. Dengan menggunakan data konsumen, bisnis tidak hanya dapat merampingkan situs web yang mereka miliki dan tunjukkan kepada konsumen, tetapi mereka juga dapat mengoptimalkan iklan dan saran produk untuk konsumen tertentu menggunakan data mereka.

Data konsumen yang berjumlah besar biasanya terdapat di toko online. Data konsumen di toko online dapat mencakup berbagai jenis informasi yang dikumpulkan dari interaksi konsumen dengan platform toko online seperti Data Informasi pribadi, Data transaksi, Data Produk dan Data Prilaku.

1. **Algoritma**

Algoritma adalah urutan instruksi atau aturan yang sistematis, terstruktur, dan terdefinisi dengan baik untuk menyelesaikan sebuah masalah atau tugas secara efektif dan efisien. Algoritma digunakan dalam berbagai bidang, termasuk matematika, ilmu komputer, teknologi informasi, fisika, dan banyak lagi [4].

Algoritma sering digunakan dalam pemrograman komputer sebagai panduan untuk menyelesaikan suatu tugas atau masalah tertentu. Algoritma dapat ditulis dalam bahasa yang mudah dimengerti oleh manusia, yang kemudian diterjemahkan ke dalam bahasa yang dimengerti oleh komputer, seperti bahasa pemrograman.

Algoritma biasanya memiliki tujuan tertentu dan dapat digunakan untuk menyelesaikan berbagai masalah, mulai dari yang sederhana hingga yang kompleks. Beberapa contoh masalah yang dapat diselesaikan dengan algoritma adalah sorting (pengurutan) data, pencarian data, enkripsi data, dan optimisasi pemrosesan data.

1. **Algoritma *String Matching***

Algoritma *string matching* adalah algoritma untuk mencari keberadaan sebuah pola atau substring dalam sebuah string. Pola atau substring tersebut dapat berupa satu karakter atau beberapa karakter yang harus ditemukan dalam sebuah string.

Algoritma string matching sangat umum digunakan dalam pengembangan perangkat lunak, seperti dalam aplikasi pencarian atau manipulasi string. Contoh penggunaan algoritma string matching adalah ketika kita ingin mencari sebuah kata tertentu dalam sebuah teks [5].

1. **Algoritma *Naive String Matching***

Naive string matching adalah salah satu algoritma string matching yang paling sederhana dan mudah dipahami. Algoritma naive string matching membandingkan setiap karakter dari sebuah pola dengan setiap karakter di dalam sebuah string. Algoritma ini memulai dengan menempatkan pola di awal string dan kemudian membandingkan karakter demi karakter. Jika karakter-karakter tidak cocok, maka pola dipindahkan satu karakter ke kanan dan proses pencocokan kembali dilakukan [6].

Algoritma naive string matching memiliki kompleksitas waktu sebesar O(mn), di mana m adalah panjang pola dan n adalah panjang string. Algoritma ini bekerja dengan baik pada string yang relatif pendek dan pola yang singkat.

1. **Algoritma *Knuth Morris Pratt***

Algoritma Knuth-Morris-Pratt (KMP) adalah metode efisien yang digunakan dalam teknik pencarian string dalam komputasi. Dikembangkan oleh Donald Knuth, Vaughan Pratt, dan James H. Morris pada tahun 1977, teknik ini mengeliminasi kebutuhan untuk mundur ke titik awal dalam string yang dicari, sehingga meningkatkan efisiensi pencarian secara signifikan.

Dalam konteks pencarian string, algoritma konvensional sering menghadapi kendala berupa berulang kali mengulang proses pencarian dari awal string setelah setiap kesalahan yang ditemukan. Ini menghasilkan efisiensi yang rendah. Algoritma KMP, sebaliknya, memindahkan titik pencarian maju setelah setiap kesalahan, sehingga mengurangi repetisi.

Algoritma KMP mengandalkan teknik bernama "preprocessing". Dalam tahap ini, algoritma membangun "fungsi pinggiran" (Border Function) dari string yang dicari, yang digunakan untuk menentukan sejauh mana kursor harus digeser jika ada ketidakcocokan.

Berikut adalah langkah-langkah operasional algoritma KMP:

1. Mulai dari kiri, bandingkan setiap karakter string yang dicari dengan string yang sedang dikombinasikan.
2. Jika ada ketidakcocokan:
   * Lihat tabel lompatan untuk karakter yang tidak cocok.
   * Geser string yang dicari sejauh yang ditentukan oleh tabel, atau geser ke kanan jika tidak ada entri di tabel.
3. Ulangi proses ini sampai string yang dicari dipindahkan sepenuhnya melalui string yang sedang dikombinasikan. [7].
4. **Algoritma *Boyer Moore***

Algoritma Boyer-Moore diciptakan oleh Bob Boyer dan J. Strother Moore pada tahun 1977. Algoritma ini mencocokkan string mulai dari akhir string kunci, bergerak menuju awalnya. Bila terdapat variasi antara karakter akhir string kunci dan string yang dicocokkan, maka setiap karakter dalam potongan string yang dicocokkan diperiksa satu demi satu. Langkah ini bertujuan untuk mengidentifikasi apakah terdapat karakter dalam potongan string tersebut yang identik dengan karakter dalam string kunci.

Jika ada kemiripan, maka string kunci akan diatur ulang sedemikian rupa sehingga memposisikan karakter yang sama secara seimbang, kemudian proses pencocokan karakter terakhir string kunci dijalankan kembali. Sementara itu, jika tidak ada karakter yang cocok, maka semua karakter string kunci akan di-shift ke arah kanan sepanjang m karakter, di mana m adalah jumlah karakter dalam string kunci. [8].

### **Mekanisme umum alogitma Boyer Moore**

Dalam penggunaan algoritma boyer moore secara umum terdiri dari dua teknik yang harus dilakukan yaitu:

**The looking-glass technique**

Teknik cermin (looking glass technique) diaplikasikan untuk menguji kesesuaian pola P terhadap teks T, dengan memulai pemeriksaan dari indeks paling akhir pada P. Proses inspeksi terhadap T tetap berasal dari permulaan, sehingga dalam situasi ini, indeks I akan dimulai pada nilai m - 1.

Contoh :

A green and blue squares with white text

Description automatically generated

Gambar 2. 1 The looking-glass technique

Dalam gambar diatas indeks **i** dimulai dari karakter **N** di Teks dan indeks **j** dimulai dari karakter **H** di Pattern.

**The character-jump technique**

Character jump technique merupakan Teknik Ketika terjadi mismatch antara (P[j] != T[i] dengan pada saat itu T[i] = x) maka algoritma ini akan melakukan lompatan atau pergeseran pada posisi tertentu. Untuk menentukan posisi lompatan ini terdapat 3 kasus yang mungkin terjadi yaitu diantaranya yaitu:

1. **Kasus 1**

Dalam kasus ini mismatch terjadi pada T[i] dan P[j] dan karakter apada T[i] yaitu x, kemudian jika terdapat karakter x di P dengan posisi indeks yang lebih kecil dari j, kemudian geser P ke kanan agar posisi x di T[i] sejajar dengan posisi kemunculan terakhir/Last Occurrence(LO) x di P.

A diagram of a diagram

Description automatically generated

Gambar 2. 2 Kasus 1 - The character-jump technique

Pada pemeriksaan selanjutnya, indeks j selalu dimulai pada posisi indeks terakhir (m-1) yang di geser adalah posisi indeks i dengan nilai i yang baru i = i + (m-1) – LO = i+m-(lo+1). Pada contoh gambar diatas i yang baru adalah = i + (4-1)-0=i+3.

1. **Kasus 2**

Dalam kasus ini mismatch terjadi pada T[i] dan P[j] dan karakter apada T[i] adalah x, kemudian jika terdapat karakter x di P tapi pada posisi indeks yang lebih besar daripada j. geser P satu karakter ke kanan agar posisi indeks terakhir P sejajar dengan posisi akhir T sebelumnya + 1.

A diagram of a computer

Description automatically generated

Gambar 2. 3 Kasus 2 - The character-jump technique

Pemeriksaan indeks berikutnya tetap dimulai dari Pattern j = m-1 kemudian yang digesera adalah nilai i yang baru adalah i = i + m – j, pada kasus dalam gambar diatas i yang baru adalah i = i + 4 – 1 = i + 3.

1. **Kasus 3**

Pada kasus ini mismatch terjadi pada T[i] dan P[j] dan karakter apada T[i] adalah x, kemudian jika tidak ditemukan karakter x pada P maka geser P agar posisi indeks pertama P sejajar dengan indeks i + 1.

A screenshot of a computer

Description automatically generated

Gambar 2. 4 Kasus 3 - The character-jump technique

Pemeriksaan indeks berikutnya tetap dimulai dari indeks terakhir Pattern j = m-1, kemudian yang digeser adalah nilai i yang baru adalah i = i + m. Pada kasus ini i yang baru adalah i = i + 4.

### **Fungsi Last Occurrence**

Fungsi last occurrence digunakan sebagai pre-procesing dalam algoritma ini. Ketiga kasus *The character Jump* memerlukan informasi mengenai di mana karakter pada Teks yang mismatch tersebut kemunculan terakhir karakternya pada Pattern.

Informasi ini dapat diproses saat Pattern sudah diketahui, fungsi ini bertugas untuk menentukan posisi kemunculan terakhir semua karakter pada Teks (T) di dalam Pattern (P). Jika karakter Teks tidak pernah muncul di Pattern maka nilai nya adalah -1. Dibawah merupakan contoh dari fungsi Last occurrence L(x).

Variasi karakter pada T:A={a,b,c,d}

A group of black letters

Description automatically generated

L(a) = 4 -> Last occurrence karakter a pada P ada di indeks 4

L(b) = 3 -> Last occurrence karakter b pada P ada di indeks 3

L(c) = 5 -> Last occurrence karakter c pada P ada di indeks 5

L(d) = -1 -> Last occurrence karakter d tidak muncul

A group of numbers in a grid

Description automatically generatedSemua parameter dari Fungsi L(x) adalah semua karakter pada T. Kemudian Semua nilai disimpan dalam tabel atau larik

1. **Algoritma *Rabin Karp***

Algoritma Rabin-Karp mewakili teknik lanjutan dalam domain pencocokan string. Dikembangkan oleh Michael O. Rabin dan Richard M. Karp, algoritma ini memberikan inovasi substansial dalam domain pencocokan string melalui pendekatan berfokus pada hashing.

Hashing dalam konsep Rabin-Karp sangat berbeda dengan strategi hashing tradisional. Hashing di sini diaplikasikan untuk masing-masing substring dalam teks yang dicocokkan. Teknik berfokus pada pembuatan 'nilai hash' yang khusus dari string kunci dan substring yang dicocokkan dalam teks.

Eksekusi algoritma Rabin-Karp melibatkan pembandingan nilai hash string kunci dengan nilai hash dari masing-masing substring dalam teks. Apabila terdapat kecocokan nilai hash, algoritma akan melanjutkan ke tahap selanjutnya yaitu pencocokan karakter demi karakter.

Keunggulan utama dari algoritma Rabin-Karp adalah efisiensi operasional. Algoritma ini secara kasar mencapai kompleksitas waktu O(n+m) dalam kasus rata-rata, di mana n dan m merujuk pada panjang teks dan string kunci secara berturut-turut. Namun, algoritma Rabin-Karp juga memiliki keterbatasan, yakni dalam kasus terburuk, algoritma ini dapat memiliki kompleksitas waktu sebesar O(nm) [9].

1. **SQL Query Like**

*SQL Query Like* adalah sebuah pernyataan (statement) yang digunakan dalam bahasa SQL untuk melakukan pencarian data yang cocok dengan pola tertentu pada sebuah tabel atau database. Dalam *SQL Query Like*, kita dapat menggunakan wildcard characters atau karakter joker seperti % (untuk merepresentasikan nol atau lebih karakter) dan \_ (untuk merepresentasikan satu karakter) untuk mencari data yang cocok dengan pola yang diinginkan [10].

Contohnya, kita dapat menggunakan pernyataan *SQL Query Like* untuk mencari semua data yang memiliki kata "apple" pada nama buah, seperti:

SELECT \* FROM fruits WHERE name LIKE '%apple%';

Pernyataan ini akan mengembalikan semua data pada tabel "fruits" yang memiliki kata "apple" pada kolom "name".

1. **UML (Unified Modelling Language)**

Untuk merancang sistem yang berorientasi objek, dibutuhkan suatu metode pemodelan secara visual, metode ini dinamakan Bahasa Pemodelan Terpadu (UML). UML adalah standar bahasa untuk pendokumentasian, perancangan, dan visualisasi.

Tujuan dibuatnya UML adalah agar mempermudah dalam mengembangkan suatu perangkat lunak. Selain itu UML juga diharapkan dapat mempermudah semua kebutuhan pengguna dengan tepat, lengkap dan efektif. merancang dan juga memodelkan sistem secara matang akan menghasilkan sistem yang baik. UML penting sekali bagi para pengembang sistem karena UML akan menjadi jembatan untuk menerjemahkan antara pengembang sistem dengan pengguna [11] Beberapa macam UML yang ada antara lain:

### ***Use Case* Diagram**

Use case diagram adalah diagram UML (Unified Modeling Language) yang digunakan dalam rekayasa perangkat lunak untuk menggambarkan interaksi antara sistem atau aplikasi dengan pengguna atau aktor-aktor lain yang terkait. Use case diagram dapat digunakan untuk menggambarkan berbagai skenario penggunaan (use case) dari sebuah sistem atau aplikasi, sehingga dapat membantu dalam memahami kebutuhan fungsional dari sistem atau aplikasi tersebut.

Dalam use case diagram, setiap use case direpresentasikan sebagai oval dan setiap aktor direpresentasikan sebagai kotak. Garis yang menghubungkan use case dan aktor menunjukkan interaksi antara mereka. Use case diagram dapat digunakan untuk mengidentifikasi dan memvisualisasikan aktor-aktor yang terlibat dalam sebuah sistem atau aplikasi, serta skenario-skenario penggunaan yang dapat terjadi.

Use case diagram juga dapat digunakan untuk mengidentifikasi dan memperjelas persyaratan sistem atau aplikasi dengan menggambarkan interaksi antara sistem atau aplikasi dan pengguna atau aktor-aktor lain yang terkait. Use case diagram sering digunakan sebagai langkah awal dalam proses pengembangan perangkat lunak, dan dapat digunakan sebagai alat komunikasi yang efektif antara pengembang perangkat lunak, klien, dan pemangku kepentingan lainnya. [12].



Gambar 1. 2 Contoh Use Case Diagram

### ***Activity Diagram***

Activity diagram adalah jenis diagram UML (Unified Modeling Language) yang digunakan untuk menggambarkan alur aktivitas atau tindakan dalam sebuah proses bisnis atau sistem. Diagram ini memperlihatkan urutan tindakan yang terjadi dalam sebuah proses atau sistem dan kondisi yang terjadi pada setiap langkah. Activity diagram sangat berguna dalam menggambarkan aliran kerja sistem, urutan aktivitas yang terjadi, serta pengambilan keputusan dalam suatu proses.

Pada activity diagram, aktivitas direpresentasikan oleh persegi panjang dengan nama aktivitas di dalamnya. Tindakan sederhana dapat direpresentasikan oleh lingkaran kecil di dalam aktivitas. Keputusan direpresentasikan oleh berlian dan memiliki beberapa jalur keluar. Sedangkan fork digunakan untuk merepresentasikan percabangan dalam sebuah proses.

Activity diagram sangat berguna dalam menggambarkan aktivitas dan tindakan dalam sebuah proses, menggambarkan kondisi yang mungkin terjadi, dan menggambarkan percabangan dalam suatu proses. Diagram ini dapat membantu pengembang perangkat lunak untuk memahami alur kerja dalam sistem, mengidentifikasi masalah dan perbaikan dalam alur kerja, serta merancang sistem yang lebih efisien dan efektif. Activity diagram sering digunakan dalam analisis dan perancangan sistem, dan dapat digunakan sebagai alat komunikasi yang efektif antara pengembang perangkat lunak, klien, dan pemangku kepentingan lainnya.



Gambar 2. 5 Contoh Activity Diagram

### ***Sequence Diagram***

Sequence diagram adalah jenis diagram UML (Unified Modeling Language) yang digunakan untuk menggambarkan interaksi antara objek dalam sebuah sistem. Diagram ini memperlihatkan urutan pemanggilan metode antara objek, serta urutan pesan yang dikirimkan antara objek dalam sebuah skenario. Sequence diagram sangat berguna dalam memodelkan interaksi antara objek dalam sistem dan memperlihatkan urutan pemanggilan metode yang terjadi.

Pada sequence diagram, objek direpresentasikan oleh sebuah kotak vertikal, dengan nama objek di bagian atas. Urutan pemanggilan metode antara objek direpresentasikan oleh panah horizontal, dengan tanda kurung kurawal di atasnya untuk menunjukkan nama metode. Pesan yang dikirim antara objek direpresentasikan oleh panah vertikal dengan tanda kurung kurawal di atasnya untuk menunjukkan nama pesan.

Sequence diagram berguna dalam menggambarkan interaksi antara objek dalam sebuah sistem, serta memperlihatkan urutan pemanggilan metode yang terjadi. Diagram ini dapat membantu pengembang perangkat lunak untuk memahami bagaimana objek dalam sistem saling berinteraksi, mengidentifikasi masalah dan perbaikan dalam interaksi antara objek, serta merancang sistem yang lebih efisien dan efektif. Sequence diagram sering digunakan dalam analisis dan perancangan sistem, dan dapat digunakan sebagai alat komunikasi yang efektif antara pengembang perangkat lunak, klien, dan pemangku kepentingan lainnya. [12].



Gambar 2. 6 Contoh Sequence Diagram

### ***Class Diagram***

Class diagram adalah jenis diagram UML (Unified Modeling Language) yang digunakan untuk memodelkan struktur kelas atau objek dalam sebuah sistem. Diagram ini memperlihatkan kelas-kelas dalam sistem, serta hubungan antara kelas-kelas tersebut. Class diagram sangat berguna dalam memodelkan struktur kelas dalam sistem dan memperlihatkan hubungan antara kelas-kelas tersebut.

Pada class diagram, kelas direpresentasikan oleh sebuah persegi panjang, dengan nama kelas di dalamnya. Atribut kelas direpresentasikan oleh nama atribut di dalam kelas, sedangkan metode kelas direpresentasikan oleh nama metode di dalam kelas. Hubungan antara kelas direpresentasikan oleh panah yang mengarah dari kelas yang satu ke kelas yang lain [12].



Gambar 2. 7 Contoh Class Diagram

1. **PHP (PHP: Hypertext Preprocessor)**

PHP adalah singkatan dari "Hypertext Preprocessor". Sejarahnya bermula pada tahun 1994, ketika Rasmus Lerdorf menciptakannya untuk mengelola statistik pribadi pada laman web-nya. Seiring perkembangan, PHP telah menjelma menjadi bahasa skrip server-side yang populer untuk pengembangan web.

PHP adalah bahasa pemrograman berbasis skrip yang ditafsirkan pada saat runtime, berarti kode PHP dievaluasi dan dijalankan di server, bukan di sisi klien. Salah satu keunikan PHP adalah bahwa kode dapat ditanamkan langsung ke dalam HTML, yang memungkinkan integrasi yang mulus antara back-end dan front-end pada aplikasi web.

Selanjutnya, PHP mendukung pemrograman berorientasi objek (OOP), memungkinkan pengembang untuk mengorganisir kode dalam bentuk kelas, objek, dan metode. Fitur ini membantu meningkatkan modularitas dan keterbacaan kode, serta memfasilitasi pemrograman skala besar.

Hingga sekarang, PHP tetap menjadi salah satu bahasa pemrograman dominan dalam bidang pengembangan web, yang dikagumi karena fleksibilitas dan kemudahannya dalam mengintegrasikan dengan HTML. Meski demikian, seperti bahasa pemrograman lainnya, PHP memiliki kekuatan dan kelemahan tersendiri, dan keefektifan penggunaannya bergantung pada konteks spesifik proyek atau aplikasi.

1. **Basis Data (MYSQL)**

MySQL dikembangkan oleh Swedish MySQL AB mereka, yang kemudian diakuisisi oleh Oracle. Pasca akuisisi, beberapa pengembang dari MySQL memutuskan untuk mencabangkan MySQL dan menciptakan MariaDB, melahirkan server basis data relasional alternatif yang mengutamakan komunitas dan pengembangan dengan sumber terbuka.

MariaDB adalah sebuah sistem manajemen basis data relasional (RDBMS) yang berfungsi untuk menyimpan, menyalurkan, dan mengelola data dalam struktur tabel. MariaDB bertujuan untuk menjaga kompatibilitas penurunan dengan MySQL, memastikan bahwa aplikasi dan perangkat lunak yang ditulis untuk MySQL tetap berjalan dengan baik di MariaDB.

Pada dasarnya, basis data MariaDB menyimpan data dalam tabel relasional, yang diorganisir dalam kolom dan baris. Karakteristik utama yang membedakan MariaDB dari MySQL termasuk paket tambahan berorientasi keamanan, peningkatan kemampuan baru, serta teknologi penyimpanan yang diperkenalkan oleh komunitas pengembang.

Salah satu kekuatan utama MariaDB terletak pada skalabilitasnya. MariaDB memiliki kemampuan untuk menangani basis data dalam berbagai ukuran, membentuk jalan yang layak diteroka bagi organisasi yang menangani pengelolaan data dalam skala besar.

Meskipun MariaDB merupakan turunan dari MySQL, namun fitur dan optimalisasi baru yang ditawarkan dapat mempengaruhi pilihan antara kedua sistem tersebut. Oleh karenanya, MariaDB telah meraih kepercayaan dan adopsi luas dalam industri, berkat pendekatannya terhadap kompatibilitas, sumber terbuka, dan partisipasi komunitas.

Database atau basis data adalah kumpulan informadi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Kegunaan utama sistem basis data adalah agar pemakai mampu menyusun suatu pandangan (view) abstraksi data.