PEMODELAN SISTEM INFORMASI LANJUTAN

Nur Yandi BYM, S.Pd., M.Pd. | Erlita, S.Pd., M.Pd.
Dr. Muh. Sabri, MH. | Dr. Sam Hermansyah, S.Pd., M.Pd.
Erwin Nurjiono | Ernita Ramadhani BYM, S.H., M.H.

PEMODELAN SISTEM INFORMASI LANJUTAN

Nur Yandi Bym,S.Pd.,M.Pd
Erlita,S.Pd.,M.Pd
Drs. Muhammad Sabri, MH
Dr. Sam Hermansyah,S.Pd.,M.Pd
Erwin Nurjiono,s.Ps.,M.Pd
Ernita Ramadhani Bym, S.H., M.H.

PEMODELAN SISTEM INFORMASI LANJUTAN

Penulis:
Nur Yandi Bym,S.Pd.,M.Pd
Erlita,S.Pd.,M.Pd
Drs. Muhammad Sabri, MH
Dr. Sam Hermansyah,S.Pd.,M.Pd
Erwin Nurjiono,s.Ps.,M.Pd
Ernita Ramadhani Bym, S.H., M.H.

Editor:
Erwin,S.Pd.,M.Pd
Adam Mudinillah,S.Pd.,M.PdLayouter :

Cover:
Rusli

Cetakan Pertama : Januari 2025

Perkumpulan Rumah Cemerlang Indonesia
ANGGOTA IKAPI JAWA BARAT
Pondok Karisma Residence Jalan Raflesia VI D.151
Panglayungan, Cipedes Tasikmalaya – 085223186009

Website : www.rcipress.rcipublisher.org
E-mail : rumahcemerlangindonesia@gmail.com

- Cet. I – : Perkumpulan Rumah Cemerlang Indonesia, 2024
; 18 x 25cm
ISBN : -

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah Swt. atas rahmat dan karunia-Nya, sehingga buku berjudul Pemodelan Sistem Informasi Lanjutan ini dapat disusun dan diselesaikan dengan baik. Buku ini hadir sebagai salah satu upaya untuk memberikan pemahaman yang lebih mendalam mengenai konsep-konsep lanjutan dalam pemodelan sistem informasi, khususnya bagi mahasiswa program studi Sistem Informasi.

Dalam era digital yang terus berkembang, pemodelan sistem informasi memegang peranan penting dalam mendukung desain, pengembangan, dan implementasi sistem yang efisien dan efektif. Buku ini dirancang untuk memberikan pemahaman mendalam tentang teknik-teknik pemodelan yang lebih kompleks, termasuk penerapan pendekatan berbasis objek, analisis proses bisnis, perancangan database, dan integrasi teknologi terbaru seperti sistem berbasis cloud dan arsitektur layanan

Kami menyadari bahwa penyusunan buku ini tidak terlepas dari dukungan dan bimbingan berbagai pihak. Oleh karena itu, kami mengucapkan terima kasih kepada para dosen pembimbing yang telah memberikan arah dan masukan berharga selama proses penyusunan. Kami memahami bahwa buku ini masih jauh dari kesempurnaan. Kritik dan saran dari pembaca sangat kami harapkan untuk perbaikan di masa yang akan datang. Semoga buku ini dapat bermanfaat bagi seluruh pembaca, khususnya mahasiswa Sistem Informasi, dalam memperluas wawasan dan meningkatkan kompetensi di bidang pemodelan sistem informasi lanjutan.

Sidenreng Rappang, 09 Oktober 2024

Penulis

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENGANTAR PEMODELAN SISTEM INFORMASI LANJUTAN

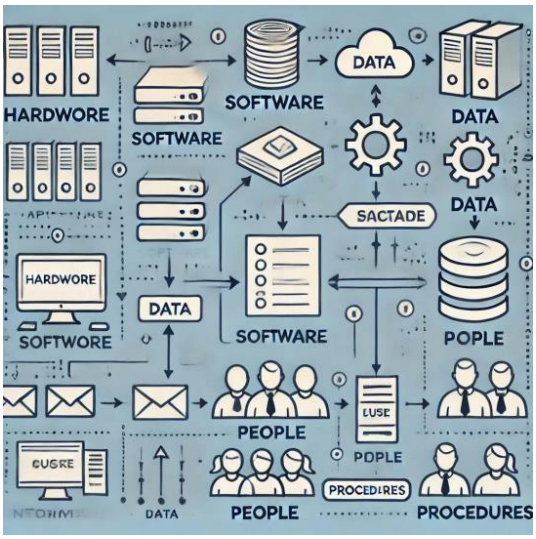
A. Konsep Dasar Sistem Informasi

Konsep dasar sistem informasi merupakan landasan penting dalam memahami bagaimana teknologi informasi dapat digunakan untuk mendukung operasi, manajemen, dan pengambilan keputusan dalam organisasi. Sistem informasi mengintegrasikan perangkat keras, perangkat lunak, data, prosedur, dan sumber daya manusia untuk mengumpulkan, memproses, menyimpan, dan menyebarkan informasi yang relevan. Memahami konsep dasar ini tidak hanya penting bagi pengembang sistem, tetapi juga bagi pengguna akhir, untuk memastikan bahwa teknologi yang diterapkan dapat memberikan nilai tambah dan mendukung tujuan strategis organisasi.

1. Definisi Sistem Informasi

Sistem informasi adalah kombinasi dari teknologi, orang, dan proses yang mengumpulkan, memproses, menyimpan, dan menyebarkan data untuk menghasilkan informasi yang berguna bagi pengambilan keputusan. Sistem informasi dapat berupa perangkat lunak, perangkat keras, dan prosedur yang bekerja sama untuk mendukung fungsi organisasi.

2. Komponen Sistem Informasi



Gambar 1 Komponen Sistem Informasi

Gambar ini menunjukkan interaksi antara perangkat keras, perangkat lunak, manusia, data, dan prosedur dalam sebuah sistem informasi. Penjelasan gambar:

- Perangkat keras: Fisik komputer dan jaringan yang menyimpan dan memproses data.
- Perangkat lunak: Aplikasi yang memungkinkan pengguna berinteraksi dengan data.
- Data: Informasi yang dikumpulkan, diolah, dan disimpan dalam sistem.
- Manusia: Pengguna sistem yang memanfaatkan informasi untuk pengambilan keputusan.
- Prosedur: Aturan yang mengatur cara sistem digunakan.

a. Hardware
   Perangkat fisik yang digunakan dalam sistem, seperti komputer, server, dan perangkat jaringan.
b. Software
   Program dan aplikasi yang digunakan untuk memproses data. Ini mencakup sistem operasi, aplikasi bisnis, dan perangkat lunak analisis data.
c. Data
   Informasi yang diolah dan disimpan dalam sistem. Data harus dikelola dengan baik agar menjadi informasi yang berguna.
d. Prosedur
   Langkah-langkah yang harus diikuti untuk mengoperasikan sistem. Prosedur ini penting untuk menjaga konsistensi dan efisiensi dalam pengolahan informasi.
e. People
   Pengguna dan pemangku kepentingan yang berinteraksi dengan sistem, termasuk manajer, analis, pengguna akhir, dan teknisi.

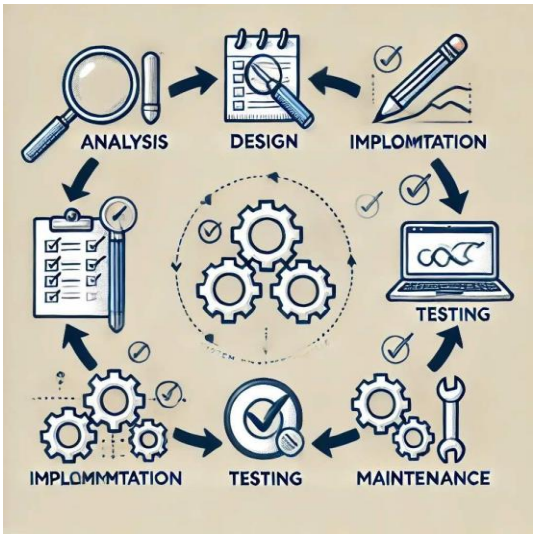
B. Pentingnya Pemodelan dalam Pengembangan Sistem

Pemodelan sistem informasi adalah proses menciptakan representasi dari sistem yang akan dibangun. Beberapa alasan mengapa pemodelan sangat penting dalam pengembangan sistem antara lain:

1. Mengurangi Kompleksitas: Dalam dunia yang semakin kompleks, pemodelan membantu menyederhanakan dan mengorganisir informasi. Dengan membuat model, pengembang dapat lebih mudah memahami struktur dan fungsi sistem yang akan dibangun.
2. Meningkatkan Komunikasi: Pemodelan menyediakan alat visual yang membantu dalam komunikasi antara tim pengembang, pengguna, dan pemangku kepentingan. Model yang jelas memungkinkan semua pihak untuk memiliki pemahaman yang sama tentang sistem yang akan dikembangkan.

3. Dokumentasi yang Baik: Model yang dibuat selama proses pengembangan berfungsi sebagai dokumentasi yang jelas dan terstruktur, yang sangat berharga untuk pemeliharaan dan pengembangan di masa mendatang.
4. Analisis Kebutuhan: Pemodelan memungkinkan analisis yang lebih mendalam terhadap kebutuhan sistem. Dengan memiliki model yang representatif, pengembang dapat mengidentifikasi dan mengatasi masalah atau kekurangan dalam spesifikasi kebutuhan.

Pemodelan memberikan cara untuk mendokumentasikan dan memahami sistem yang sedang dikembangkan. Pemodelan membantu mengurangi kompleksitas, mendeteksi kesalahan sejak dini, serta memfasilitasi komunikasi antara pengembang dan stakeholder.



Gambar 2 Siklus Pengembangan Sistem dan Pemodelan

Gambar ini menggambarkan proses pengembangan sistem informasi mulai dari analisis kebutuhan, perancangan, implementasi, hingga pengujian dan pemeliharaan, dengan pemodelan sebagai bagian dari setiap langkah. Penjelasan gambar:

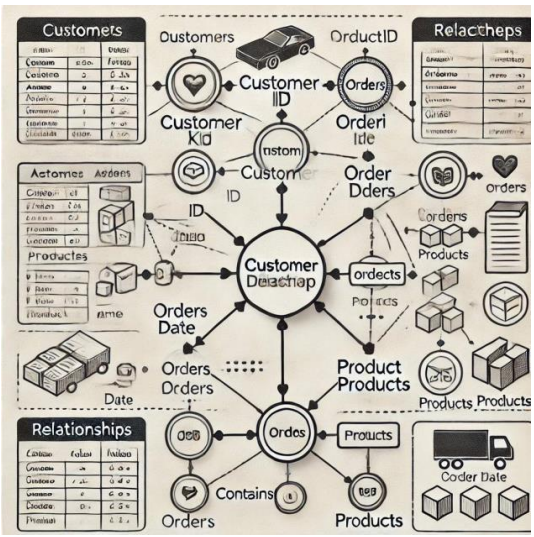
- Analisis: Mengumpulkan kebutuhan pengguna dan menganalisis sistem yang ada.
- Pemodelan: Merancang model yang menggambarkan bagaimana sistem akan bekerja.
- Implementasi: Membangun sistem berdasarkan model.
- Pengujian: Memastikan sistem bekerja sesuai dengan spesifikasi model.
- Pemeliharaan: Memperbaiki dan memperbarui sistem berdasarkan perubahan kebutuhan.

C. Perbedaan Pemodelan Dasar dan Lanjutan

Pemodelan dasar dan lanjutan dalam konteks sistem informasi atau teknologi memiliki perbedaan mendasar. Pemodelan sistem infomasi dapat dibagi menjadi dua kategori: pemodelan dasar dan pemodelan lanjutan.

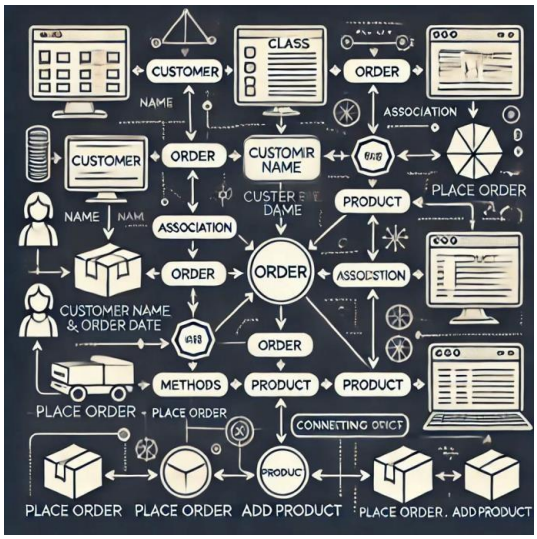
Tabel 1Perbandingan Pemodelan Dasar dan Lanjutan

Aspek	Pemodelan Dasar	Pemodelan Lanjutan
Tingkat Detail	Rendah, fokus pada elemen utama	Tinggi, mencakup detail teknis dan perilaku sistem
Skala Aplikasi	Sistem kecil dan menengah	Sistem besar dan kompleks
Tujuan	Memahami konsep dasar dan struktur umum	Optimasi, simulasi perilaku, dan analisis dinamis
Contoh Teknik	Entity-Relationship Diagram (ERD), Flowchart	Unified Modeling Language (UML), Simulasi Berbasis Agen



Gambar 3 Entity-Relationship Diagram (ERD)

Berikut adalah gambar Entity-Relationship Diagram (ERD) yang menunjukkan hubungan antara entitas seperti Pelanggan, Pesanan, dan Produk.



Gambar 4 UML Class Diagram

Berikut adalah gambar UML Class Diagram yang menunjukkan kelas-kelas seperti Pelanggan, Pesanan, dan Produk, beserta atribut dan metodenya.

D. Teknik-teknik Pemodelan Lanjutan

Beberapa teknik pemodelan lanjutan yang umum digunakan dalam pengembangan sistem informasi adalah pendekatan yang lebih mendalam dan kompleks untuk merancang dan menganalisis sistem. Teknik-teknik ini bertujuan untuk menggambarkan interaksi antar elemen sistem dengan lebih rinci, memperhatikan aspek teknis, dinamis, dan interdependensi antara berbagai komponen. Dengan menggunakan pemodelan lanjutan, pengembang dapat memastikan bahwa sistem yang dibangun lebih efisien, scalable, dan mampu mengakomodasi kebutuhan yang lebih kompleks. Beberapa teknik pemodelan lanjutan yang umum digunakan dalam pengembangan sistem informasi adalah:

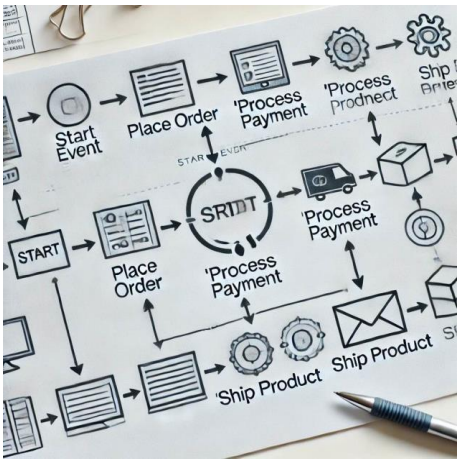
1. Unified Modeling Language (UML)

UML adalah bahasa pemodelan standar yang digunakan untuk menggambarkan sistem perangkat lunak. UML mencakup berbagai jenis diagram yang digunakan untuk menggambarkan berbagai aspek sistem:

a. Diagram Kasus Penggunaan: Menyediakan gambaran tentang bagaimana pengguna berinteraksi dengan sistem.
b. Diagram Kelas: Menggambarkan struktur sistem dengan menampilkan kelas, atribut, metode, dan hubungan antar kelas.
c. Diagram Aktivitas: Menggambarkan alur kerja dalam suatu proses atau interaksi.

2. Business Process Model and Notation (BPMN)

BPMN adalah notasi standar yang digunakan untuk memodelkan proses bisnis. Ini memungkinkan organisasi untuk menggambarkan proses secara jelas, sehingga semua pemangku kepentingan dapat memahami alur kerja dan interaksi yang terjadi.

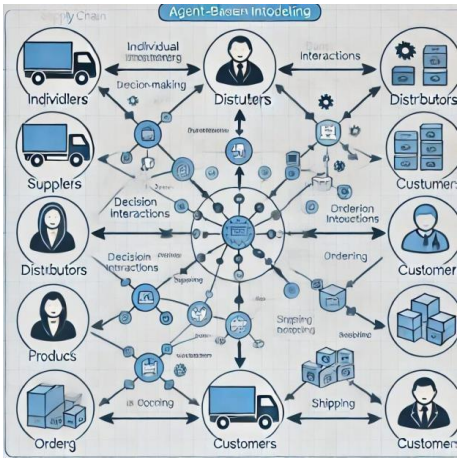


Gambar 5 Business Process Model and Notation (BPMN)

Berikut adalah gambar Business Process Model and Notation (BPMN) yang menggambarkan alur proses pemesanan, mulai dari pesanan ditempatkan, diproses, hingga pengiriman produk.

3. Modeling Architecture

Teknik ini mencakup pemodelan arsitektur perangkat lunak, yang bertujuan untuk menggambarkan struktur, komponen, dan interaksi antar komponen dalam sistem. Pemodelan arsitektur membantu dalam memahami bagaimana sistem berfungsi secara keseluruhan.



Gambar 6 Agent-Based Modeling (ABM)

Berikut adalah gambar Agent-Based Modeling (ABM) yang menggambarkan interaksi antara agen-agen seperti pemasok, distributor, dan pelanggan dalam rantai pasokan.

E. Tantangan dalam Pemodelan Sistem Informasi

Meskipun pemodelan sistem informasi memiliki banyak manfaat, seperti meningkatkan pemahaman tentang kebutuhan pengguna dan mempermudah perancangan sistem yang efisien, ada beberapa tantangan yang perlu dihadapi dalam proses pemodelan ini. Tantangan-tantangan tersebut sering kali terkait dengan kompleksitas sistem yang harus dimodelkan, kesulitan dalam menggambarkan interaksi dinamis antar komponen, serta kebutuhan untuk terus memperbarui dan menyesuaikan model dengan perkembangan teknologi dan perubahan kebutuhan bisnis, ada beberapa tantangan yang perlu dihadapi:

1. Keterbatasan Komunikasi

Meskipun pemodelan dapat meningkatkan komunikasi, perbedaan pemahaman antara pemangku kepentingan dapat menjadi hambatan. Untuk mengatasi hal ini, penting untuk melibatkan semua pihak dalam proses pemodelan dan memastikan bahwa model yang dihasilkan dapat dipahami oleh semua.

2. Dinamika Kebutuhan

Kebutuhan sistem dapat berubah seiring waktu, memerlukan model yang fleksibel dan mudah diubah. Pengembang harus siap untuk beradaptasi dan memperbarui model sesuai dengan perubahan kebutuhan.

3. Integrasi dengan Teknologi Baru

Dengan cepatnya perkembangan teknologi, pemodelan harus mampu beradaptasi dengan alat dan teknik baru yang muncul. Pemodelan yang baik harus mempertimbangkan inovasi terbaru dan bagaimana hal itu dapat diterapkan dalam sistem yang sedang dikembangkan.

BAB II
METODOLOGI PEMODELAN LANJUTAN

A. Unified Modeling Language (UML) Tingkat Lanjut

UML (unified modeling language)merupakan bahasa pemodelan visual yang digunakan untuk mendesain dan mendokumentasi sistem perangkat lunak. UML membantu dalam memvisualisasikan, membangun, dan mendokumentasikan sistem perangkat lunak, serta membantu dalam komunikasi antara pengembang, analisis, dan pemangku kepentingan lainnya.

1. Diagram Kelas Lanjutan

Diagram kelas merupakan diagram UML yang digunakan untuk menggambarkan struktur dan hubungan antar kelas dalam sistem perangkat lunak. Diagram kelas tingkat lanjutan memperkenalkan konsep-konsep yang lebih kompleks untuk memodelkan hubungan antar kelas dengan lebih detail.

a. Asosiasi
   Asosiasi adalah hubungan antara dua kelas yang menunjukkan bahwa objek dari kedua kelas tersebut dpat saling berinteraksi. Asosiasi dapat diwakili dengan garis yang menghubungankan kedua kelas, dengan panah yang menunjukkan arah navigasi.
b. Agregasi
   Agregasi adalah jenis asosiasi khusus yang menunjukkan hubungnan “has-a” (memiliki)antara dua kelas. Kelas agregat (kelas yang memiliki) memiliki hubungan “has-a” denagan kelas aggregasi (kelas yang dimiliki). Agregrasi diwakili dengan garis berlian yang berhubung ke kelas agregat.
c. Komposisi
   Komposisi adalah jenis agregrasi yang lebih kuat, menunjukkan hubungan “partof” (bagian dari) antara dua kelas. Kelas komposisi (kelas yang memiliki) memiliki kontrol yang penuh atas kelas yang dikomposisikan (kelas yang dimilik). Komposisi diwakili dengan garis berlian yang terhubung ke kelas komposisi dengan ujung yang diisi.
d. Generalisasi
   Generalisasi adalah hubungan “is-a”(adalah) antara dua kelas. Kelas anak (subclass) mewarisi atribut dan meode dari kelas induk (superclass). Generalisasi diwakili dengan garis segitiga yang terhubung dengan kelas induk.

2. Diagram Aktivitas Komples

Diagram aktivitas adalah diagram UML yang digunakan untuk menggambarkan alut kerja atau proses dalam sistem . diagram aktivitas tingkat lanjutan memperkenalkan konsep-konsep yang lebih kompleks untuk memodelkan alur kerja yang rumit.

a. Aktivitas
   Aktivitas adalah tindakan atau operasi yang dilakukan dalam alur kerja. Aktivtas diwakili dengan bentuk persegi panjang dengan sudut tumpul
b. Transisi
   Transisi adalah alitan kontrol yang menghubungan dua aktivitas. Transisi diwakili dengan garis panah yang menguhubungkan dua aktivitas.
c. Kondisi
   Kondisi adalah ekspresi boolwan yang menentukan apakah transisi akan dieksekusi. Kondisi diwakili dengan bentuk berlian yang terhubung ke transisi.
d. Fork Dan Join
   Fork dan join adalah dapat digunakan untuk memodelkan proses paralel. Fork digunakan untuk membagi alur kerja memjadi beberapa alur paralel, sedangkan join digunakan untuk menggambungkan kembali alur paralel.

3. Diagram Komponen Dan Penyebaran diagram komponen UML

Diagram Komponen Dan Penyebaran diagram komponen UML memberikan gambaran konseptual tentang interaksi antara berbagai sistem. Diagram ini menunjukkan bagaimana komponen- komponen perangkat lunak saling berhubungan dan bagaiamana mereka berkomunikasi satu sama lain

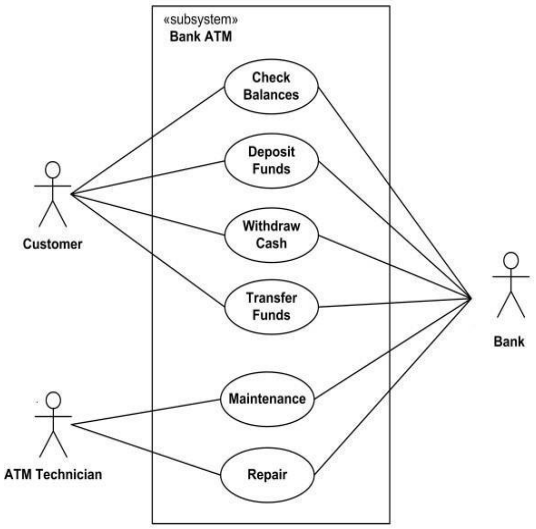
a. Komponen
   Komponen adalah unit perangkat lunak yang independen yang menyediakan fungsionalitas tertentu dalam sistem. Komponen dapat berupa modul, library, atau layanan web.
b. Antarmuka
   Antarmuka adalah spesifik yang mendefinisikan bagaimana komponen dapat diakses dan digunakan oleh komponen lain.
c. Ketergantungan
   Ketergantungan menunjukkan bahwa satu komponen bergantung pada komponen lain untuk berfungsi. Ketergantungan diwakili dengan garis putus-putus yang menghubungkan dua komponen.

d. Keuntungan diagram komponen

- Memahami arsitektur sistem : diagram komponen membantu dalam memahami struktur fisik sistem perangkat lunak dan bagaimana komponen-komponennya saling berhubungan.
- Identifikasi ketergantungan : diagram komponen membantu dalam mengidentifikasi ketergantungan antara komponen- komponen, yang dapat membantu dalam mengelola perybahan dan mengurangi resiko.
- Pengembangan modular : diagram komponen mendukung pengembangan modular, di mana komponen dapat sikembangkan dan diuji secara independen.

4. Contoh Diagram UML yang Sering Digunakan

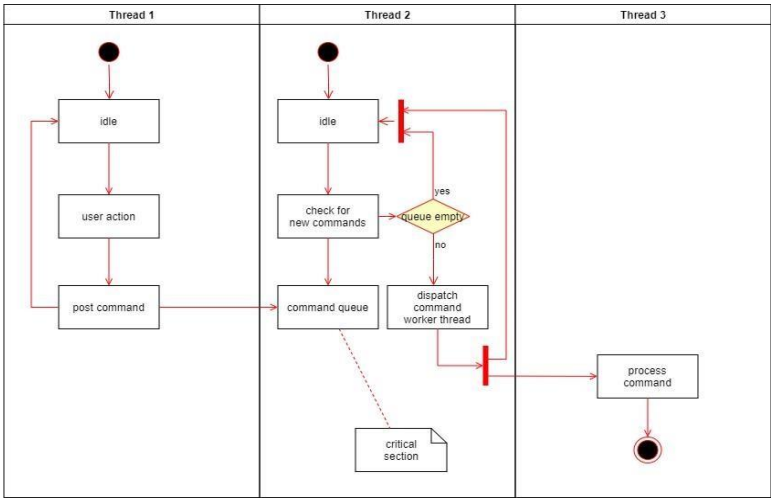
a. Use Case Diagram



Gambar 7 *(Use Case diagram ATM)*

Use Case Diagram adalah satu jenis dari diagram UML (Unified Modelling Language) yang menggambarkan hubungan interaksi antara sistem dan aktor. Use Case dapat mendeskripsikan tipe interaksi antara si pengguna sistem dengan sistemnya. Use Case merupakan sesuatu yang mudah dipelajari. Langkah awal untuk melakukan pemodelan perlu adanya suatu diagram yang mampu menjabarkan aksi aktor dengan aksi dalam sistem itu sendiri, seperti yang terdapat pada Use Case.

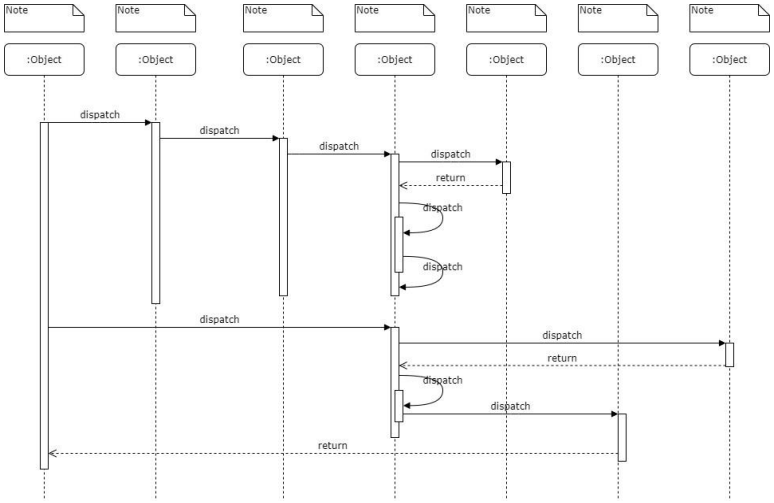
b. Activity Diagram



Gambar 8 *(Activity Diagram)*

Activity diagram atau dalam bahasa Indonesia berarti diagram aktivitas, merupakan sebuah diagram yang dapat memodelkan berbagai proses yang tejadi pada sistem. Seperti layaknya runtutan proses berjalannya suatu sistem dan digambarkan secara vertikal. *Activity* diagram adalah salah satu contoh diagram dari UML dalam pengembangan dari *Use Case*.

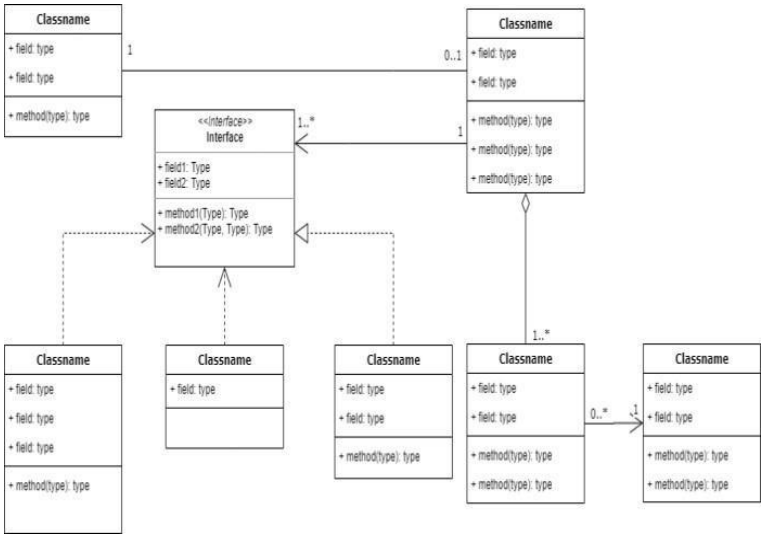
c. Sequence Diagram



Gambar 9 *(Sequence Diagram)*

Sequence diagram merupakan diagram yang menjelaskan interaksi objek berdasarkan urutan waktu. *Sequence* dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu, seperti yang tertera pada Use Case diagram.

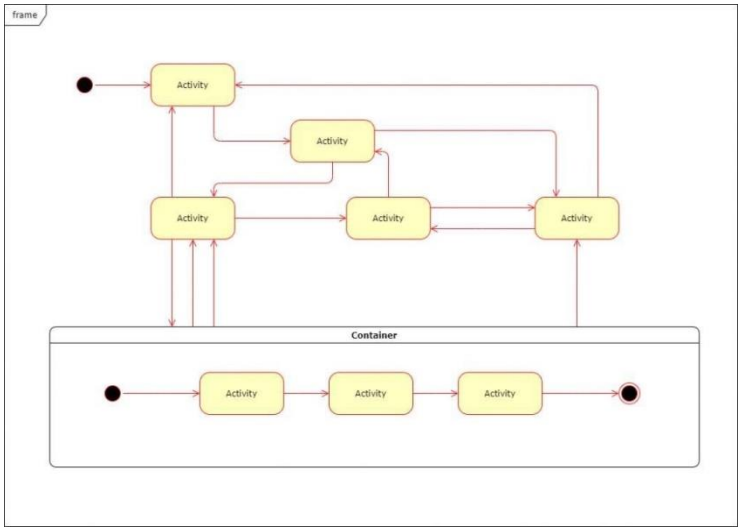
d. Class Diagram



Gambar 10 *(Class Diagram)*

Class diagram atau diagram kelas merupakan suatu diagram yang digunakan untuk menampilkan kelas-kelas berupa pake-paket untuk memenuhi salah satu kebutuhan paket yang akan digunakan nantinya. Namun, pada Class diagram desain modelnya dibagi menjadi 2 bagian. Class diagram yang pertama merupakan penjabaran dari domain model yang merupakan abstraksi dari basis data. Class diagram yang kedua merupakan bagian dari modul program MVC pattern (Model View Controller), di mana terdapat class boundary sebagai class *interface*, *class control* sebagai tempat ditemukannya algoritma, dan *class entity* sebagai tabel dalam basis data dan *query* program.

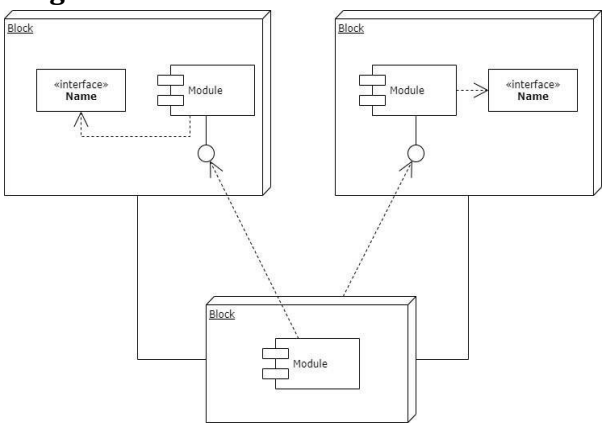
e. Statemachine Diagram



Gambar 11 *(Statemachine Diagram)*

Statemachine yaitu salah satu jenis diagram pada UML yang berfungsi untuk menggambarkan transisi serta perubahan pada suatu objek pada sistem.

f. Component Diagram



Gambar 12 *(Component Diagram)*

Component diagram yang berfungsi untuk menggambarkan *software* pada suatu sistem. *Component* diagram merupakan penerapan pada piranti lunak atau *software* dari satu class maupun lebih, dan biasanya berupa *file* data, *source code,.exe, table*, dokumen, atau yang lainnya.

B. Business Procces Model and Notation (BPMN)

Business procces model and notation (BPMN) adalah standar industri untuk memodelkan proses bisnis. BPMN membantu dalam memvasualisasikan, menganlisis, dan mengoptimalkan proses bisnis dengan menggunakan diagram yang mudah dipahami oleh berbagai pemangku kepentingan.

1. Konsep dasar BPMN

- Proses bisnis : serangkaian aktivitas yang terstruktur yang menghasilkan nilai bagi pelanggan.
- Pemodelan proses bisnis : membuat representasi visual dari proses bisnis untuk memahami, menganilisis, dan mengoptimalkannya.
- BPMN : standar industri untuk memodelkan proses bisnis dengan menggunakan diadram yang mudah dipahami.

2. Elemen BPMN

Diagram BPMN terdiri dari berbagai elemen yang mewakilkan berbagai aspek proses bisnis, seperti :

- Event: kejadian yang memicu atau mengakhiri aktivitas.
- Activity : tindakan atau tugas yang dilakukan dalam proses.
- Gateway : titik keputusan dalam proses.
- Flow : aliran kontrol yang menghubungkan elemen-elemen dalam proses

- Pool : representasi dari entitas organisasi yang terlibat dalam proses.
- Lane : representasi dari peran atau fungsi dalam intetias organisasi

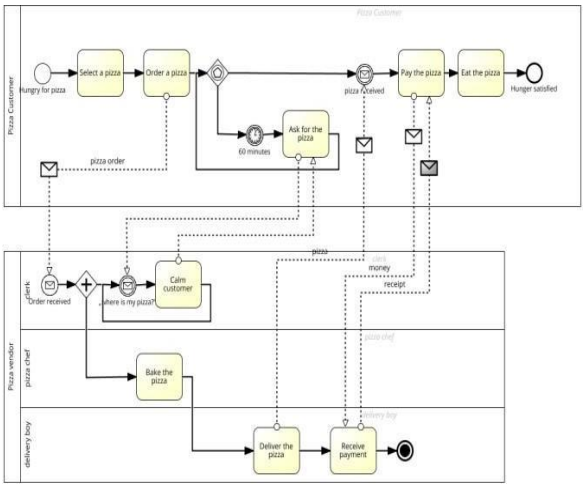
3. Jenis diagram BPMN

Terdapat beberapa jenis diagram BPMN antara lain:

- Diagram proses : diagram utamayang menggambarkan keseluruhan proses bisnis.
- Diagram kolaborasi : diagram yang menunjukkan interkasi antara berbagai entitas organisasi dalam proses.
- Diagram event : diagram yang fokus pada event yang terjadi dalam proses.

4. Contoh BPMN

- Layanan Pizza



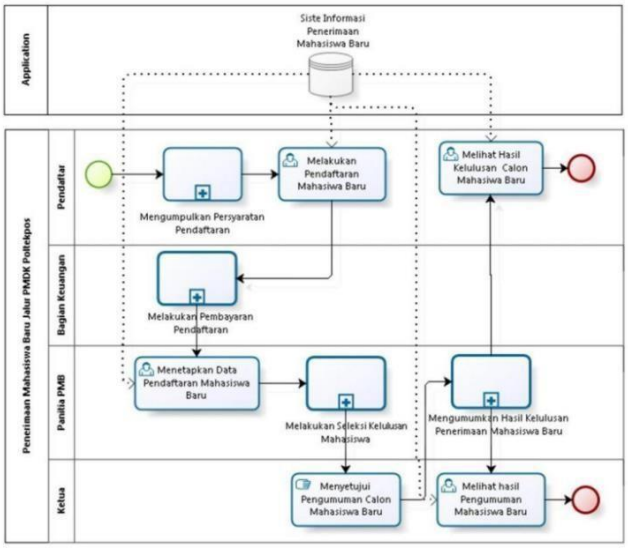
Gambar 13 *(Layanan Pizza)*

Penggambaran BPMN antara pelanggan pizza dan vendor pizza yang menunjukkan alur proses yang terjadi pada pemesanan pizza hingga pembayaran pesanan pizza oleh pelanggan pizza kepada vendor. Penggambaran dilakukan dengan diagram kolaborasi.

- Pendaftaran Mahasiswa

Keterangan :

a. Simbol Database yang disebut dengan artifak yang menunjukan bahwa data yang ada masuk kedalam sistem
b. Simbol bulat berwarna hijau menandakan awal proses dan merah sebagai akhir proses
c. Tanda (+)plus pada kotak proses menunjukan didalam proses tersebut masih ada proses lain di dalamnya.



Gambar 14 *(Pendaftaran Mahasiswa)*

Pemodelan lanjutan mengacu pada penggunaan teknik statistik, matematika, dan komputasi yang canggih untuk membangun model yang kompleks dan akurat. Model-model ini dapat digunakan untuk emprediksi, menganalisis, dan mengoptimalkan sistem yang kompleks.

Beberapa konsep dasar dalam pemodelan lanjutan meliputi:

- Data : data merupakan bahan baku utama dalam pemodelan lanjutan. Kualits dan kuantitas data sangat memengaruhi akurasi model.
- Algoritma : algoritma adalah serangkaian intruksi yang digunakan untuk membangun dan melatih model.
- Model : model adalah representasi matematis atau statistik dari sistemm yang ingin dipelajari.
- Evaluasi : evaluasi model dilakukan intuk mengukur kkinerja model dan menentukan apakah model tersebut sesuai dengan tujuan yang ingin dicapai.
- Representasi sistem : metodologi pemodelan lanjutan bertujuan untuk membangun model yang merepresentasikan isstem nyata dengan akurat. Model ini dapat berupa persamaan matematis, diagram, atau algoritma yang menggambarkan hubungan antara berbagai variabel dalam sistem.
- Iterasi dan penempurnaan : metodologi pemodelan lanjutan bersifat iteratif. Model yang dibangun akan terus disempurnakan berdasarkan hasil evaluasi dan data baruyang tersedia. Proses ini melibatkan penguangan langkah-langkah pemodelan hingga model mencapai kinerja yang diharapkan.

Metodologi pemodelan lanjutan merupakan pendekatan yang kuat untuk membangun model yang kompleks dan akurat. Memahami konsep dasar

metodologi ini dapatmembantu kita untuk memanfaatkan kekuatan pemodelan untuk memecahkan masalah kkompleks dan membuat keputusan yang lebih baik. Pentingnya metodologi pemodelan lanjutan:

a. Pemahaman sistem yang komples
   Metodologi pemodelan lanjutan memungkinkan kita untuk memahami sistem yang kompleks dengan cara yang lebih mendalalm.
b. Prediksi yang akurat
   Model yang diangun dengan menggunakan mmetododlogi pemodelan lanjutan dapat digunakan untuk mebuat prediksi yang lebih akurat tentang perilaku sistem di masa depan
c. Pengambilan keputusan yang lebih baik
   Model yang akurat dapat memmbantu kita dalam membuat keputusan yang lebih baik,, baik dalam bisnis, ilmu pengetahuan, atau bidang lainnya.
d. Optimasi proses
   Metodologi pemodelan lanjutan dapat digunakan untuk mengoptimalkan proses dan meningkatkan efesiensi sistem.

C. Domain-Specific Languages (DSL)

Domain-specific languages (DSL) adalah bahasa pemrogramanyang di rancang khusus untuk menyelesaikan dalam domain tertentu. Berbeda dengan bahasa pemrogram umum seperti java atau python, DSL fokus pada kebutuhan spesifik suatu domain, seperti pemodelan keuangan, desain web, atau kontrol robot.

1. Konsep dasar DSL

- Domain : area atau bidang tertentu yang memiliki masalah dan kebutuhan spesifik.
- Bahasa : sistem formal untuk mengekspresikan ide dan intruksi.
- DSL : bahasa yang dirancang khusus untuk menyelesaikan masalah dalam domain tertentu.

2. Jenis DSL

- Internal DSL : DSL yang dibangun di atas bahasa pemrograman umum, biasanya menggunakan library atau framework.
- Eksternal DSL : DSL yang memiliki sintaks dan semnatik sendiri, bisanya diimplementasikan sebagai bahasa pemrograman kecil atau alat khusus.

3. Keuntungan DSL

- Kode yang lebih ringkas : DSL memungkinkan pengembangan untuk mengekspresikan solusi dengan cara yang lebih ringkas dan mudah dipahami dalam domain tertentu.
- Peningkatan produktivitas : dengan fokus pada domain tertentu, DSL meningkatkan produktivitas pengembang dan mengurangi kesalahan.
- Kemudahan pemeliharaan : kode DSL lebih mudah dipahami dan diubah, sehingga memudahkan pemeliharaan sistem.
- Meningkatkan kolaborasi : DSL memungkinkan kaloborasi yang lebih baik antara pengembang dan pengguna domain, karena bahasa yang digunakan lebih mudah dipahami.

4. Contoh DSL

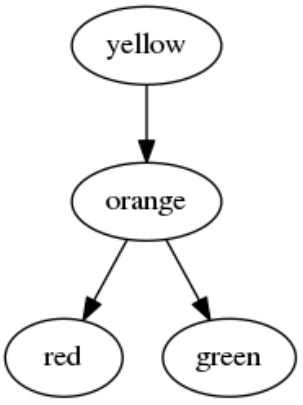
a. DOT – DSL Untuk Mendefinisikan Grafik

DOT merupakan bahasa yang dapat menggambarkan grafik, baik yang berarah maupun tidak berarah.

```
Digraf Nama Graf {
Kuning - > Oranye - > Merah;

Oranye - > Hijau;
```

Dari uraian ini, gambar yang mewakili grafik tersebut dapat dibuat. Untuk melakukannya, Anda menggunakan program bernama graphviz, yang bekerja dengan bahasa DOT. Dari contoh sebelumnya, Anda akan mendapatkan ini:



Gambar 15 (Yang Dihasilkan Menggunakan Dot DSL)

Bahasa ini juga memungkinkan untuk menentukan bentuk simpul, warnanya, dan banyak karakteristik lainnya. Namun, dasar- dasarnya cukup sederhana dan hampir semua orang dapat mempelajari cara menggunakannya dalam hitungan menit.

b. PlantUML – DSL Untuk Menggambar Diagram UML

PlantUML Dapat digunakan untuk mendefinisikan diagram UML dari berbagai jenis. Misalnya, kita dapat mendefinisikan diagram sekuens.

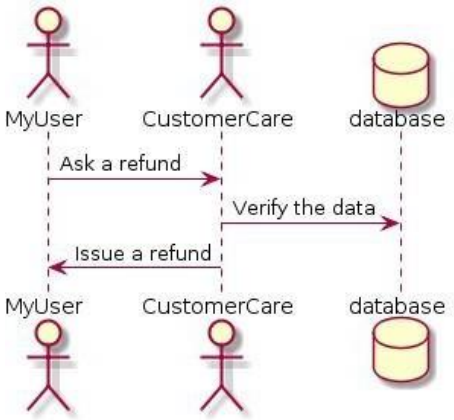
```
@startuml      aktor MyUser  aktor CustomerCare  basis data

basis data MyUser - > CustomerCare : Ajukan pengembalian

dana

CustomerCare - > database : Verifikasi data
```

Dari definisi ini kita dapat memperoleh gambaran.



Gambar 16 (Yang Dihasilkan Dengan PlantUML DSL)

```
panjang ( $ 0 ) > 80

Atau hitung baris dalam        sebuah

berkas: AKHIR { cetak NR }
```

Dengan sintaksis yang serupa, berbagai jenis diagram dapat didefinisikan seperti diagram kelasatau diagram kasus penggunaan. Menggunakan DSL tekstual untuk mendefinisikan diagram UML memiliki beberapa keuntungan: lebih mudah untuk diversikan, dan dapat dimodifikasi oleh semua orang tanpa alat khusus. DSL seperti ini dapat digunakan selama rapat untuk mendukung diskusi: saat peserta berdebat, berbagai diagram dapat didefinisikan dengan cepat dan gambar yang sesuai dibuat dengan sekali klik.

c. Sed – DSL untuk mendefinisikan transformasi teks

Pada sistem operasi mirip UNIX (Linux, Mac, BSD, dll.) terdapat seperangkat alat baris perintah, yang masing-masing menerima instruksi dalam formatnya sendiri. Format ini dapat dianggap sebagai DSL yang memungkinkan untuk menentukan tugas yang akan dieksekusi. Misalnya *sed* mengeksekusi transformasi teks yang ditunjukkan menggunakan DSL-nya sendiri.

```
Apakah Anda ingin mengganti kata “Jack” dengan kata “John”?

s/Jack/John/g


Atau apakah Anda ingin menghapus semua baris file dari baris

10 hingga kata “stophere” ditemukan? 10,/stoper/hari
```

Tingkat teknis yang diperlukan untuk menjadi ahli dengan DSL jenis ini sudah tinggi. Namun, banyak pengguna komputer tingkat lanjut dapat mempelajari dasardasar untuk menjalankan operasi umum pada berkas. Hal-hal kecil yang perlu mereka lakukan setiap hari dan yang saat ini dilakukan secara manual. Misalnya, Anda dapat memiliki templat email yang berisi placeholder seperti “{FIRST_NAME}” dan menggantinya dengan pengujian yang tepat dengan salah satu perintah ini.

d. Gawk – DSL Untuk Mencetak Dan Memproses Teks

Seperti sed, gawk adalah utilitas UNIX lain yang menerima perintah dalam bahasanya sendiri. Misalnya, Anda dapat mencetak semua baris dari file tertentu yang panjangnya lebih dari 80 karakter:

Filosofi UNIX adalah menggunakan beberapa utilitas kecil ini dan menggabungkannya untuk melakukan tugas yang paling menakjubkan dan rumit. Misalnya, Anda dapat mengambil file input, mengubahnya dengan *sed* , lalu mencetak bagian yang dipilih menggunakan *gawk* .

e. Gherkin – DSL Untuk Mendefinisikan Tes Fungsional Ketimunadalah DSL untuk mendefinisikan pengujian fungsional.

Sintaksnya sangat fleksibel sehingga tampak seperti teks bebas. Pada dasarnya, pengembang, analis, dan klien dapat duduk bersama dan mendefinisikan beberapa skenario. Skenario ini kemudian dapat dieksekusi sebagai pengujian, untuk memverifikasi apakah aplikasi memenuhi harapan. Berikut ini adalah cara kita dapat mendefinisikan harapan untuk penarikan uang dari ATM:

```
Skenario: Verifikasi penarikan di ATM berfungsi dengan benar

Mengingat John memiliki $ 500 di akunnya


Ketika John meminta untuk menarik 200 $

Dan John memasukkan PIN yang benar

Kemudian 200 $ dikeluarkan melalui ATM
```

f. Website-Spec – DSL Untuk Pengujian Web Fungsional

```
Buka $url
Jam pada pembuatan

# Pilih toko

Di dalam toko panel kartu
Pilih ` [ tanggal-uji=menyimpan ] label`
```

Gherkin bukan satu-satunya DSL yang digunakan untuk mendefinisikan pengujian. Spesifikasi situs web_dapat digunakan untuk menentukan pengujian fungsional khusus untuk aplikasi web. Di sini, kami menentukan cara menavigasi situs web tertentu dan apa yang kami harapkan untuk ditemukan.

```
Ingat tes sebagai $StoreName
Klik
Pilih tombol lanjutkan

! Kelas tidak boleh berisi "disabled" Klik
Pilih elemen `.preview-value`
Teks properti harus $StoreName
```

Dalam kasus ini, pengembang tidak perlu mendefinisikan terjemahan perintah ke GPL karena bahasa ini menggunakan perintah khusus domain, seperti "Klik", yang dapat dijalankan oleh penerjemah. Dengan pelatihan minimal, kini siapa pun dapat menjelaskan interaksi spesifik dengan situs web dan hasil yang diharapkan.

g. SQL – Basis Data

Anda mungkin pernah mendengar tentang SQL. Ini adalah bahasa yang digunakan untuk menentukan cara memasukkan, mengubah, atau mengekstrak data dari basis data relasional. Mari kita lihat beberapa statistik dari tabel *STATS*:

```
PILIH MAKS ( SUHU_F ) , MIN ( SUHU_F ) , RATA-RATA ( HUJAN_I ) , ID
DARI STATISTIK
KELOMPOKKAN BERDASARKAN ID;
```

Tentu saja Anda tidak mengharapkan orang biasa dapat menulis kueri yang rumit: SQL bukanlah bahasa yang mudah dan memerlukan waktu untuk dikuasai. Namun, Anda tidak perlu dilatih sebagai pengembang untuk mempelajari SQL.

Memang banyak DBA yang bukan pengembang. Mungkin Joe tidak boleh dipercaya untuk menulis akses ke basis data, tetapi ia dapat memperoleh akses baca dan menulis kueri sederhana untuk menjawab pertanyaannya sendiri alih-alih harus bertanya kepada seseorang dan menunggu untuk mendapatkan jawaban.

Misalkan ia perlu mengetahui suhu maksimum di bulan Agustus di Atlanta:

```
PILIH MAKS ( nilai ) DARI SUHU DI MANA kota= "Atlanta" DAN bulan=
```

Mungkin Joe tidak akan pernah mencapai level DBA, tetapi ia dapat mempelajari beberapa pertanyaan dasar dan menyesuaikannya dengan kebutuhannya, membuatnya lebih mandiri dan membiarkan rekan kerjanya fokus pada pekerjaan mereka alih-alih membantunya.

h. HTML – Tata Letak Web

Saya sangat berharap Anda pernah mendengar tentang bahasa yang cukup berhasil ini untuk mendefinisikan dokumen. Sungguh menakjubkan untuk berpikir bahwa kita dapat mendefinisikan halaman HTML 20 tahun yang lalu, ketika kebanyakan orang memiliki komputer desktop yang terhubung ke monitor dengan resolusi 640x480 piksel dan sekarang beberapa halaman tersebut dapat ditampilkan pada browser yang berjalan di ponsel pintar kita. Saya kira ini adalah contoh yang baik tentang apa yang dapat dicapai dengan DSL.