

## BAB 2

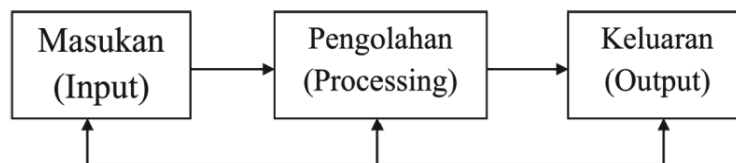
### TINJAUAN PUSTAKA

#### 2.1 Konsep Sistem Informasi

##### 2.1.1 Sistem

Sistem adalah suatu komponen atau jaringan prosedur yang saling bergantung dan bekerja sama membentuk suatu jaringan untuk mencapai tujuan atau sasaran tertentu. Menurut yang lain, sistem adalah suatu jaringan dari proses – proses yang saling berhubungan dan bergabung bersama untuk melakukan suatu kegiatan atau mencapai suatu tujuan tertentu[1][2].

Sedangkan unsur-unsur yang mewakili suatu sistem pada umumnya berupa masukan (*input*), proses (*process*), dan keluaran (*output*). Hal ini dapat dijelaskan dalam model sistem berikut:



**Gambar 2.1** Model Sistem[1]

Gambar diatas menunjukkan bahwa suatu sistem atau pendekatan sistem setidaknya harus mempunyai empat komponen yakni masukan, pengolahan, keluaran[1].

Dari kedua pandangan diatas, sistem dapat dipahami sebagai kumpulan subsistem, komponen ataupun elemen yang saling bekerja sama untuk tujuan yang sama untuk menghasilkan hasil yang telah ditentukan sebelumnya[2].

### **2.1.2 Informasi**

Informasi adalah data yang telah diolah untuk individu, organisasi atau siapapun yang membutuhkannya. Informasi akan berguna jika penerimanya meminta informasi tersebut[2].

Informasi berguna bagi pengambil keputusan karena mengurangi ketidakpastian (atau meningkatkan pengetahuan). Informasi sangatlah penting karena berdasarkan informasi, manajer dapat mengetahui situasi obyektif perusahaan. Informasi tersebut merupakan hasil pengolahan data atau peristiwa yang dikumpulkan dengan metode atau cara tertentu[1].

### **2.1.3 Sistem Informasi Kasir**

Sistem informasi kasir adalah sistem informasi yang mengatur serangkaian prosedur dan metode untuk membuat, menganalisis, dan mengumpulkan informasi yang memudahkan transaksi bagi pemilik usaha, kasir, dan pelanggan, seperti pengolahan data persediaan, data penjualan, dan data vendor. Dan mengelola keuangan perusahaan menggunakan laporan bisnis yang dihasilkan oleh sistem[3].

## **2.2 HTML**

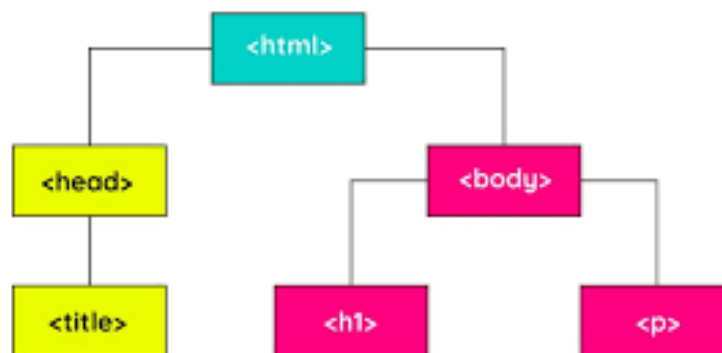
Hypertext Markup Language (HTML) adalah bahasa standar yang digunakan untuk menampilkan konten pada halaman web. Fungsi yang dapat dilakukan dengan menggunakan bahasa pemrograman HTML adalah:

1. Mengatur dan mendesain konten halaman di website,
2. Membuat tabel pada halaman website,

3. Mempublikasikan situs website secara online,
4. Membuat form yang dapat digunakan sebagai input dan proses registrasi dan transaksi melalui website,
5. Menampilkan area gambar di browser.

Tanda perintah dalam bahasa pemrograman Hypertext Markup Language (HTML) disebut dengan tag. Tag digunakan untuk mendefinisikan tampilan dokumen HTML. Tag HTML menunjukkan bahwa isi file tersebut adalah dokumen. Elemen head merupakan kepala dari dokumen HTML. Elemen head digunakan untuk mengatur identitas file, sedangkan tag body digunakan untuk mendefinisikan konten yang akan ditampilkan pada halaman web[4].

Struktur pemrograman HTML dapat dilihat pada gambar berikut :



**Gambar 2.2** Struktur Pemrograman HTML

## **2.3 Javascript**

*JavaScript* adalah bahasa pemrograman berbentuk kumpulan script yang berjalan pada suatu dokumen HTML. *JavaScript* dapat menyempurnakan tampilan dan sistem pada halaman web-based application yang dikembangkan.

Adapun karakteristik dari bahasa pemrograman *JavaScript* adalah:

1. Bahasa pemrograman berjenis high-level programming
2. Bersifat client-side
3. Berorientasi pada objek
4. Bersifat loosely typed

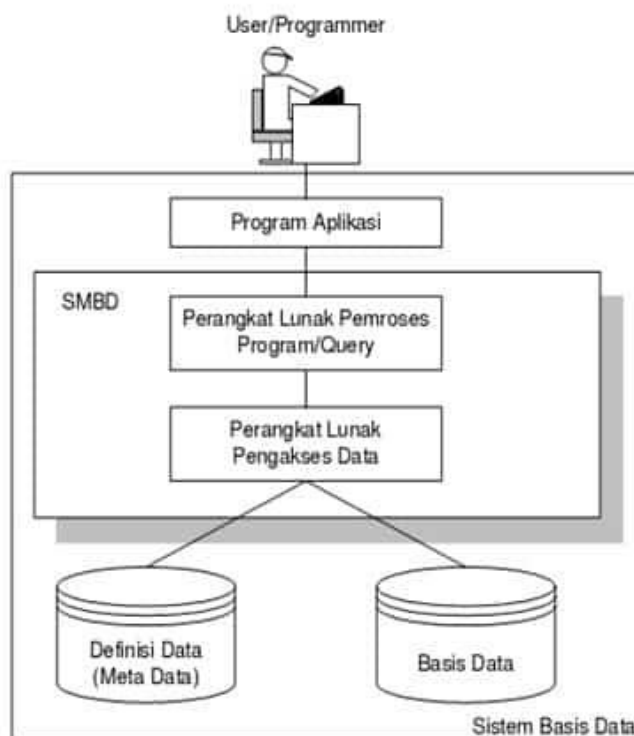
Aplikasi atau tools standar yang digunakan dalam mengimplementasikan *JavaScript* adalah perangkat lunak pengedit teks seperti Notepad++, Adobe Dreamweaver dan NetBeans. Sedangkan web browser yang digunakan seperti: Mozilla Firefox, Google Chrome, Internet Explorer, Opera Mini, Safari, dan lain sebagainya. Kelemahan dari bahasa pemrograman *JavaScript* ini sendiri adalah *JavaScript* tidak didukung oleh browser versi lama[4].

## **2.4 Sistem Basis Data**

### **2.4.1 Definisi Sistem Basis Data**

Basis Data terdiri dari kata basis dan data. Basis dapat diartikan sebagai markas atau gudang. Sementara itu data merupakan kumpulan fakta yang bertujuan untuk memberikan gambaran secara menyeluruh mengenai suatu keadaan. Melalui data memungkinkan seseorang dapat menganalisis, menggambarkan atau menjelaskan suatu keadaan,

Basis data (*database*) dapat dibayangkan atau digambarkan sebagai sebuah almari arsip. Jika kita memiliki sebuah almari arsip dan bertugas mengelolanya, maka kemungkinan besar kita akan melakukan hal – hal seperti: memberi map pada kumpulan arsip, memberi penomoran dengan pola tertentu yang nilainya unik pada setiap map, lalu menempatkan arsip tersebut dengan urutan tertentu didalam lemari. Walaupun hal – hal tersebut tidak seluruhnya dilakukan paling tidak semua lemari arsip menerapkan suatu aturan tertentu bagaimana keseluruhan arsip – arsip tersebut disusun[5].



**Gambar 2.3** Konsep Sistem Basis Data[5]

### **2.4.2 Prinsip dan Tujuan Basis Data**

Prinsip basis data adalah pengaturan suatu arsip atau data. Tujuan utamanya yakni memberikan kemudahan dalam pengambilan kembali data atau arsip yang pernah disimpan. perbedaan basis data dengan lemari arsip hanya terletak pada media penyimpanan. Basis data menggunakan media penyimpanan elektronik[5].

### **2.4.3 SQLite**

SQLite adalah sistem manajemen basis data relasional yang dibangun ke dalam sebuah library bahasa pemrograman C. Lite didalam SQLite dimaksudkan agar ringan dalam hal instalasi, administrasi basis data, dan sumber daya yang diperlukan. SQLite memungkinkan satu koneksi database untuk mengakses beberapa files database secara bersamaan. Oleh karena itu, ia menyediakan banyak fitur menarik seperti menggabungkan tabel pada yang database berbeda atau menyalin data antar database dengan satu perintah[6].

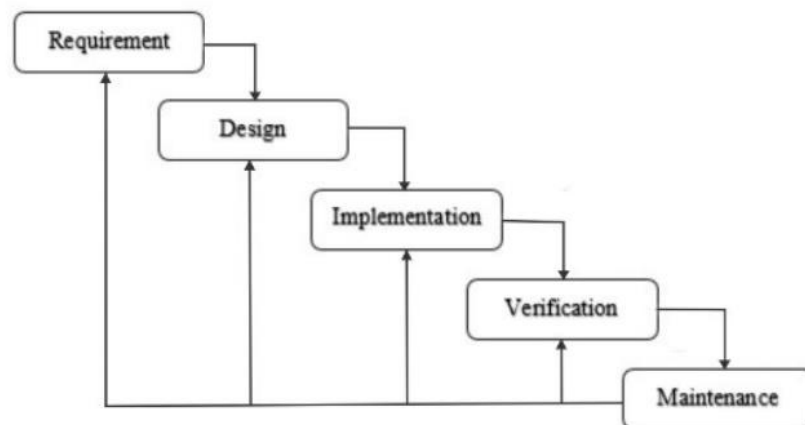
### **2.5 Metode Waterfall**

Waterfall atau air terjun adalah model yang digunakan untuk pengembangan perangkat lunak ini. Model waterfall adalah model sekuensial linier atau alur hidup klasik". Model waterfall menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau berurutan dimulai dari analisis, desain, pengkodean, pengujian dan tahap pendukung[7].

Model berkembang secara sistematis dari satu tahap ke tahap lain dalam mode seperti air terjun. Model mengusulkan sebuah pendekatan kepada

pengembangan software yang sistematis dan sekuensial yang mulai dari tingkat kemajuan sistem pada seluruh analisis, desain, kode, pengujian dan pemeliharaan.

Model pengembangan ini bersifat linear dari tahap awal pengembangan sistem yaitu tahap perencanaan sampai tahap akhir pengembangan sistem yaitu tahap pemeliharaan. Tahapan berikutnya tidak akan dilaksanakan sebelum tahapan sebelumnya selesai dilaksanakan dan tidak bisa kembali atau mengulang ke tahap sebelumnya[8].



**Gambar 2.4** Tahapan Metode Waterfall[8]

Penjelasan metode waterfall dalam penelitian ini diantaranya :

1. Requirement : Dimulai dari penyusunan latar belakang masalah yang akan diteliti merumuskan masalah beserta batasan-batasan masalah serta meneliti objek penelitian.
2. Design : Dilanjutkan dengan perancangan program menggunakan UML sesuai hasil dari observasi dan kebutuhan perusahaan.
3. Implementation : Fase ini program mulai dicoba untuk di temukan kelebihan dan kekurangannya.

4. Verification : Setelah ditemukan kekurangan dari program ini barulah di verifikasi untuk di lakukan perbaikan.
5. Maintenance : Setelah di verifikasi baru program di perbaiki sesuai dengan kebutuhan perusahaan yang terbaru[8].

## **2.6 Perangkat Pemodelan Sistem**


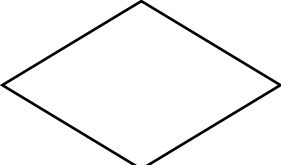
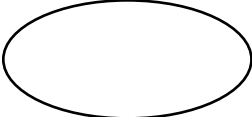

### **2.6.1 Entity Relationship Diagram (ERD)**

*Entity Relationship Diagram* (ERD) adalah suatu model jaringan kerja yang menguraikan susunan data yang disimpan dari sistem secara abstrak dan sebagai penerjemah hubungan antara entity-entity yang ada. ERD berisikan komponen-komponen entitas dan himpunan yang saling berelasi yang ditandai dengan adanya *primary key* pada masing-masing file dapat dilihat pada gambar dibawah ini

*Entity Relationship Model* digunakan untuk mengubah data yang ada di dunia nyata dalam bentuk simbolik, seperti perangkat konseptual, menjadi diagram data yang dikenal sebagai Entity Relationship Diagram[9]. Pada dasarnya ada empat simbol yang ada pada ERD. Anda bisa melihat gambar berikut untuk lebih jelasnya.



**Tabel 1** Simbol ERD

Simbol	Keterangan
	Entitas, yaitu Kumpulan dari objek yang dapat diidentifikasi secara unik
	Relasi, yaitu hubungan terjadi antara salah satu lebih entitas. Jenis hubungan antara lain, one to one, one to many dan many to many
	Atribut, yaitu karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas
	Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya.

### 2.6.2 Unified Modelling Language (UML)

Seiring berkembangnya teknologi perangkat lunak, diperlukan bahasa untuk memodelkan perangkat lunak yang dihasilkan, dan standarisasi diperlukan agar orang-orang di berbagai negara memahami pemodelan perangkat lunak. Banyak orang menciptakan bahasa pemodelan untuk pengembangan perangkat lunak berdasarkan teknologi pemrograman yang berkembang saat itu. Misalnya Data Flow Diagram (DFD) yang dikembangkan dan digunakan oleh banyak pihak digunakan untuk memodelkan perangkat lunak dengan menggunakan proses atau pemrograman structural, lalu ada juga State Transition Diagram (STD) yang digunakan untuk memodelkan sistem waktu nyata (real-time)[8].

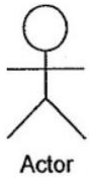
*Unified Modeling Language (UML)* adalah bahasa standar industri untuk visualisasi, desain, dan dokumentasi sistem perangkat lunak. UML menyediakan standar untuk merancang model suatu sistem.



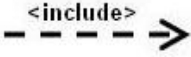
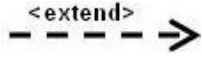



Menggunakan UML kita dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi ini dapat berjalan di perangkat keras, sistem operasi, dan jaringan apa pun dan ditulis dalam bahasa apa pun dari pemrograman. Namun, karena UML juga menggunakan kelas dan operasi matematika dalam konsep dasarnya, UML paling cocok untuk menulis perangkat lunak dalam bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Berikut diagram yang sering digunakan dalam *Unified Modeling Language (UML)* yaitu :



**a. Use Case Diagram**

Use case diagram adalah sarana untuk menggambarkan persyaratan sebuah sistem apa yang seharusnya digunakan. Use case diagram digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut. Use case memiliki simbol yang digunakan dalam percanggihannya, untuk lebih jelasnya dapat dilihat pada table dibawah ini[10].

**Tabel 2** Simbol Use Case[10]

No	Simbol	Nama	Keterangan
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan Ketika berinteraksi dengan use case.

2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen tidak mandiri (independent).
3		Generalization	Hubungan dimana objek anak (descendant) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (ancestor).
4		Include	Menspesifikasikan bahwa use case sumber secara eksplisit.
5		Extend	Menspesifikasikan bahwa use case target memperluas perilaku dari use case sumber pada suatu titik yang diberikan.
6		Association	Suatu garis yang menghubungkan suatu objek dengan objek lainnya.
7		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		Use Case	Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.



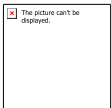
9		Collaboration	Merupakan berbagai aturan dan elemen yang bekerja guna menyediakan action yang lebih besar dari jumlah elemennya.
10		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya terkomputasi.

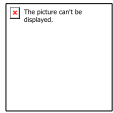
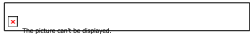

### b. Activity Diagram

Activity diagram adalah bentuk visual dari alir kerja yang berisi tindakan, pilihan, pengulangan, *concurrency*. Diagram ini dibuat untuk menjelaskan aktivitas komputer atau alur aktivitas dalam organisasi. Diagram ini menggambarkan alur kontrol secara garis besar.

Activity diagram ini memiliki komponen dengan bentuk tertentu, dihubungkan dengan tanda panah yang mengarahkan urutan aktivitas yang terjadi dari awal sampai akhir. Untuk lebih jelas ada pada table dibawah ini[10].

**Tabel 3** Simbol Activity Diagram

No	Simbol	Nama	Keterangan
1		Status Awal	Sebuah diagram aktivitas memiliki sebuah status awal.
2		Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja.
3		Percabangan / Decision	Percabangan dimana ada pilihan aktivitas yang lebih dari satu.

4		Status Akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
5		Control Flow	Control flow menjelaskan aliran kontrol dari satu tindakan ke tindakan lain
6		Object Flow	Object flow mewakili arah objek yang bergerak melalui aktivitas.

*Unified Modeling Language (UML)* biasa digunakan untuk:

- a. *Use case* dan aktor digunakan untuk menggambarkan batasan sistem dan fitur sistem secara umum.
- b. Menjelaskan aktivitas dan proses bisnis yang biasa dilakukan yang dibuat dengan diagram interaksi.
- c. Menggambarkan representasi struktur statis suatu sistem dalam bentuk diagram kelas.
- d. Gunakan diagram keadaan untuk membuat model perilaku yang menggambarkan perilaku atau properti dari suatu sistem.
- e. Jelaskan arsitektur implementasi aktual menggunakan diagram komponen dan diagram evolusi.
- f. Gunakan stereotip untuk menyampaikan atau memperluas fungsionalitas.

UML adalah salah satu alat paling terpercaya di bidang pengembangan sistem berorientasi objek. Ini karena UML menyediakan bahasa pemodelan visual

yang memungkinkan pengembang sistem membuat cetak biru untuk visi mereka dalam bentuk standar. UML bertindak sebagai jembatan dalam komunikasi beberapa aspek sistem melalui sejumlah elemen grafis yang dapat digabungkan menjadi diagram. UML memiliki banyak diagram yang dapat mengakomodasi pandangan yang berbeda dari perangkat lunak yang dibangun[9].


### 2.6.3 Data Flow Diagram (DFD)

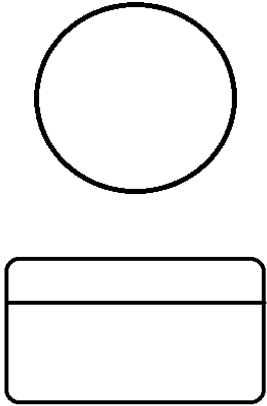
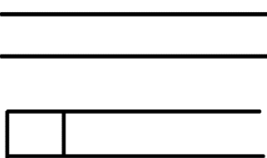
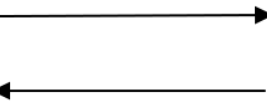

Data Flow Diagram (DFD) atau Diagram Arus Data (DAD) merupakan logika data yang digunakan untuk proses data, darimana data berasal dan tujuan dari data tersebut keluar. DFD dapat menggambarkan tujuan data keluar, dimana data tersimpan, dan proses yang diterapkan pada data tersebut, DFD digunakan untuk menggambarkan suatu aplikasi baru yang akan dikembangkan dan menggambarkan bagaimana aliran data ditransformasikan[11][12].

Data Flow Diagram (DFD) merupakan bagan yang menjelaskan mengenai arus data dari suatu perusahaan, yang digambarkan dari sejumlah simbol tertentu yang memiliki makna perpindahan data yang terjadi dalam suatu sistem[11].

Simbol – simbol DFD yang digunakan antara lain yaitu sebagai berikut :

**Tabel 4** Simbol DFD

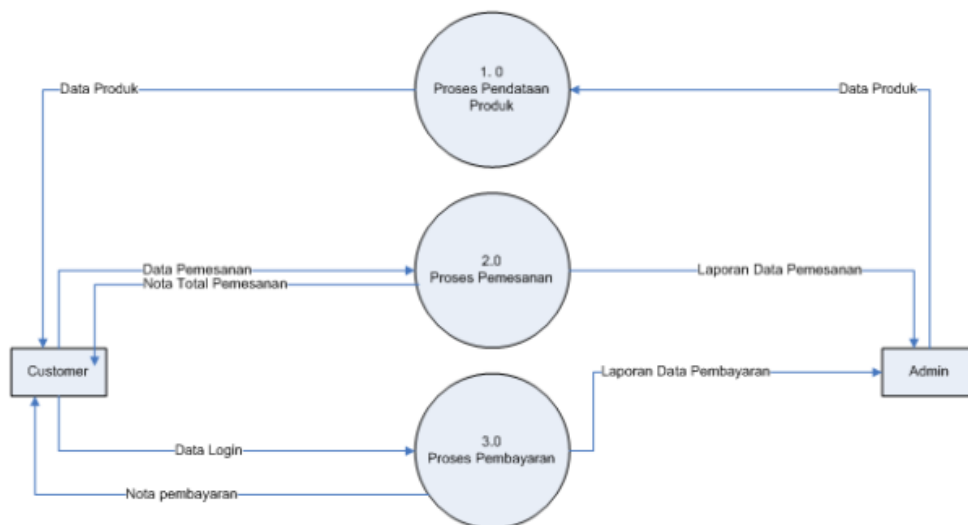
No	Nama	Simbol	Keterangan
1		External Entity	Simbol yang bisa berupa organisasi, orang, sekelompok orang, maupun Perusahaan yang sama tetapi diluar kendali oleh sistem yang tengah dibuat olehnya.

2		Process	Sebuah proses yaitu aktivitas maupun kerja yang akan dilakukan mesin, komputer maupun orang dan nantinya akan menghasilkan sebuah arus data yang akan masuk ke dalam suatu proses untuk dilakukan data yang akan keluar dari proses tersebut.
3		Data Store	Data store umumnya sangat berkaitan dengan storage atau penyimpanan. Data store juga diberikan nama yang sesuai dengan nama dari file penyimpanannya.
4		Data Flow	Data Flow juga sering disebut dengan arus data, yang mana sering disimbolkan dengan tanda panah. Arus data tersebut juga mengalir diantara suatu proses, data store, dan juga terminator. Kegunaan arus data juga untuk menunjukkan suatu arus data yang bisa berupa masukkan untuk sebuah sistem maupun hasil dari suatu proses sebuah sistem.
5		Input / Output	Simbol merupakan representasi dari input maupun output yang terdapat pada Data Flow Diagram

Jenis – jenis DFD yang sering digunakan

### 1. DFD Level 0

Jenis yang pertama ini sering disebut dengan diagram konteks. Pada DFD ini merupakan diagram yang terdiri dari metode yang dapat menjelaskan secara umum lingkup sistem informasi yang akan dibuat.

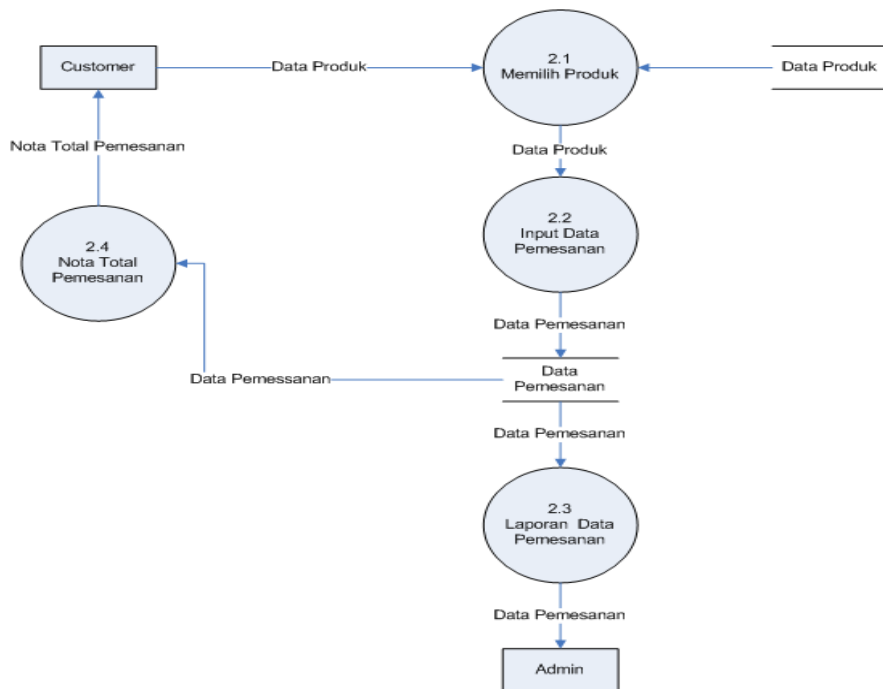


**Gambar 2.5** Contoh DFD Level 0

### 2. DFD Level 1

Jenis DFD yang kedua ini merupakan lanjutan dari diagram konteks. Pada DFD digambarkan secara lebih rinci dan lengkap karena proses utama dipecah menjadi beberapa sub dengan fungsi masing – masing.

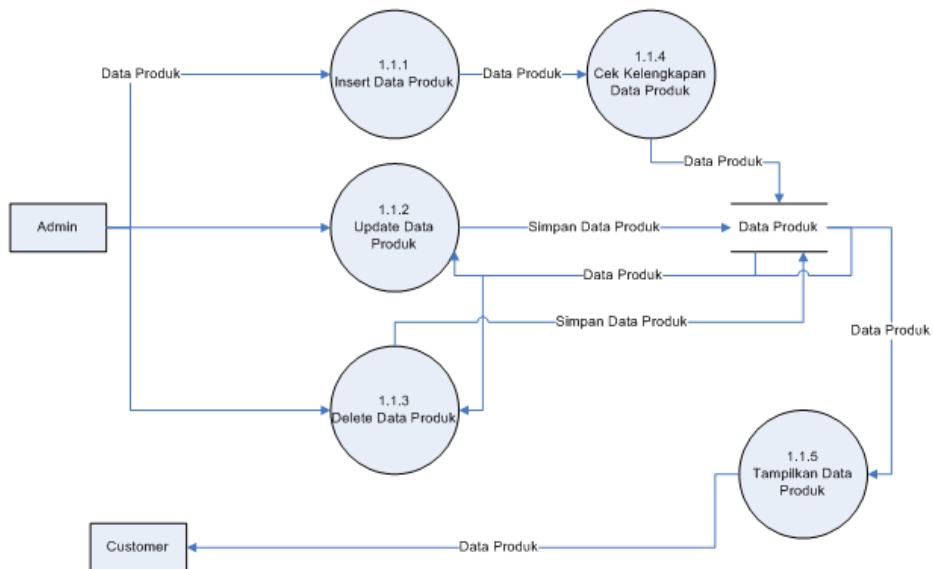




**Gambar 2.6** Contoh DFD Level 1

### 3. DFD Level 2

DFD ini akan lebih lengkap lagi mengulas proses – proses yang ada dalam lingkup sebuah sistem informasi.



**Gambar 2.7** Contoh DFD Level 2

## 2.7 Referensi Terkait

Terdapat beberapa penelitian yang berkaitan dengan penelitian dalam laporan ini, penelitian tersebut terlampir pada tabel dibawah ini

**Tabel 5** Penelitian Terdahulu

No	Penulis / Author	Judul Peneliti	Hasil
1	Rudi Hermawan dan Ahmad Fauzi (2021)	PERANCANGAN SISTEM INFORMASI KASIR PENJUALAN BARANG BERBASIS WEBSITE METODE SPIRAL TOKO WARNA	Perancangan sistem informasi kasir penjualan barang toko serba ada pada toko warna melakukan transaksi, pengolahan data, dan pencetakan laporan yang secara otomatis berbasis web apabila berjalannya akan terjadi perubahan sistem yang tadinya manual menjadi terkomputerisasi.
2	Alexander Waworuntu dan Ester Lumba (2017)	PENGEMBANGAN APLIKASI KASIR DAN PENGELOLAAN STOK BERBASIS WEB STUDI KASUS: TOKO XYZ	Penelitian ini menghasilkan sebuah aplikasi kasir dan pengelolaan stok berbasis web yang responsif. Pengguna terdiri dari tiga kelompok, yaitu pemilik, kepala toko dan kasir. Pemilik dapat mengakses semua fungsional sistem yang meliputi pengelolaan produk, penjualan, pembelian, operasional, kategori produk, metode pembayaran, pelanggan, supplier, karyawan, pengguna sistem dan laporan.

3	Yoga Handoko, Ayu Latifah dan Andi Fikri Nugraha (2021)	PERANCANGAN SISTEM INFORMASI APLIKASI KASIR PADA KAFE RESTORASI KOPI BERBASIS WEB	Pada aplikasi sistem informasi kasir ini menghasilkan pengelolaan laporan penjualan kafe menjadi lebih baik karena adanya proses yang terkomputerisasi, maka data dapat tersimpan dengan aman. Sistem informasi ini memakai database MySQL dan dikembangkan menggunakan bahasa pemrograman PHP.
4	Titania Grawidi Yurita dan Fitri Marisa (2017)	PERANCANGAN APLIKASI POINT OF SALES (POS) BERBASIS WEB MENGGUNAKAN METODE SIKLUS HIDUP PENGEMBANGAN SISTEM	Aplikasi ini dikembangkan dengan menggunakan bahasa pemrograman PHP dan database MySql. Hasil dari penelitian ini adalah sebuah aplikasi terstruktur yang dirancang untuk mengolah data penjualan.

Dengan memanfaatkan landasan teori ini, pengembangan Sistem Informasi Kasir Berbasis Desktop Untuk Manajemen Stok Dan Pelaporan Penualan Di Toko Berkah Nida dapat lebih terarah, relevan, dan mendukung tujuan serta manfaat yang diinginkan.

## 2.8 Hipotesis

Dugaan sementara bahwa sistem informasi kasir ini dapat membantu dalam penjualan di Toko Berkah Nida seperti mempermudah dalam pengelolaan stok dan pelaporan penjualan.