4.2. Perancangan Sistem

Sebelum perancangan terlebih dahulu dilakukan perencanaan untuk melakukan pengembangan sistem. Perencanaan sistem ini menyangkut estimasi dari kebutuhan-kebutuhan fisik dan tenaga kerja yang dibutuhkan untuk mendukung pengembangan sistem dan perancangan sistem dapat diartikan sebagai berikut :

1. Tahap setelah analisis dari siklus pengembangan sistem
2. Pendefinisian atas kebutuhan fungsional
3. Persiapan untuk rancang dan bangun implementasi
4. Menggambarkan sistem yang akan dibentuk
5. Penggabungan komponen hardware dan software sistem

4.2.1. Tujuan Perancangan

Perancangan sistem ini bertujuan untuk memberikan gambaran secara umum dan terperici kepada pengguna tentang sistem yang akan dibangun dan mengidentifikasikan komponen-komponen sistem informasi yang akan dirancang secara detail. Tujuan perancangan sistem informasi penjualan ini akan diuraikan sebagai berikut :

1. Meringankan kerja karyawan dengan media yang telah terkomputerisasi.
2. Meningkatkan pelayanan tehadap konsumen dan memberikan kemudahan dalam proses pengolahan data dan taransaksi.
3. Adanya Laporan penjualan yang terinci sehingga dapat mengetahui hasil yang diterima dan dapat sebagai acuan untuk kedepannya dan laporan yang ada telah tersimpan dengan media komputerisasi.

Gambaran Umum Sistem Yang Diusulkan

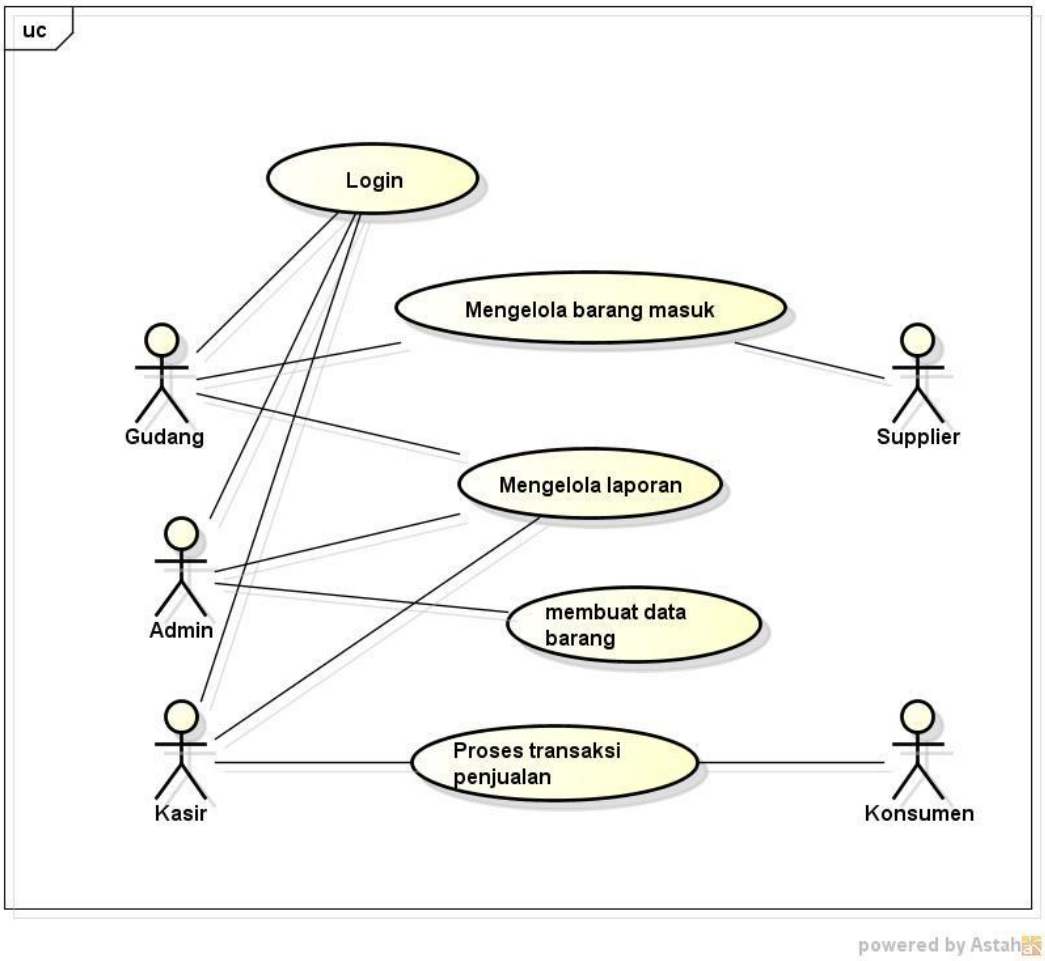
Perancangan aplikasi pembantu dan pendukung kinerja karyawan ini menggunakan bahasa UML (*Unified Modeling Language*). Mulai dari pembuatan rancangan *Usecase* nya hingga *Deployment* Diagramnya. Secara umum, proses tersebut dimulai dari penentuan arsitektur utama dari sistem yang ingin dirancang dan dibuat diagram alur proses pendaftaran sebagai Admin (Administrator/Operator) yang mana dalam kasus ini adalah *Project Manager* itu sendiri dan dilanjutkan alur proses utamanya, yang berupa pengolahan modul kerja, pengolahan informasi klien, perhitungan upah lembur hingga pengolahan absensi karyawan.

4.2.2. Perancangan Prosedur Yang Diusulkan

Usulan atau rancangan dari analisis sistem yang akan dikembangkan adalah sebagai berikut :

4.2.2.1. *Usecase Diagram*

Usecase diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *Usecase* mempresentasikan sebuah interaksi antara Aktor dengan sistem. Berikut adalah perancangan *Usecase* nya:



Gambar 4.5. *Use Case Diagram* yang di Usulkan di Naura Fashion

4.2.2.2. Sekenario Use Case

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Berikut skenario use case yang diusulkan :

1. Nama Use Case : *Login User*

   Aktor Utama : Kasir, Gudang dan Admin

   Aktor Tambahan : -

   Tujuan : Proses Validasi Akun *User*

Tabel 4.6. Skenario *Use Case Login User* yang di Usulkan

User	Sistem
1. Masuk kehalaman Login	
2. Mengisi *Username* dan *Password* pada menu Login	
	3. Verifikasi *Username* dan *password* pada sistem database.
	4. Jika account terdaftar, maka akan secara otomatis mengakses menu utama, jika account tidak terdaftar maka sistem akan memberikan peringatan *"Maaf Username atau Password yang anda masukan salah"* dan kembali ke halaman *login user*.

2. Nama Use Case : Proses *Transaksi*

   Aktor Utama : *Kasir*

   Aktor Tambahan : Konsumen

   Tujuan : Proses melakukan Transaksi

Tabel 4.7. Skenario *Proses Transaksi* yang di Usulkam

Konsumen	Kasir	Sistem
1. Memilih Barang		
	2. Bagian kasir Menerima permintaan barang	

	3. Bagian kasir melakukan proses trasaksi dengan masuk ke menu penjualan	
		4. Sistem akan menampilkan menu penjualan, no penjualan, dan tanggal penjualan
	5. Kasir mengkik tambah data	
		6. Sistem akan masuk ke menu penjualan
	7. Kasir memilih kategori dan kemudian nama barang	
		8. Sistem akan menampilakan kode barang,ukuran,harga dan stok barang.
		9. jika barang yang dipesan melebihi stok yang ada maka sistem akan menampilkan “jumlah barang dan stok barang yang tersedia hanya :…” dan jika barang yang dipesan kurang dari stok barang makan sistem akan

		memproses masuk dalam data pesanan”
	10. tekan tombol tambah untuk menyimpan kedalam keranjang belanja, dan dapat menambah barang lagi dengan kode barang yang berbeda.	
		11. Sistem akan menyimpan data barang ke daftar belanja dan menampilkan total belanja yang harus dibayar.
	12. Menambahkan diskon atau langsung menginputkan total uang yang dibayarkan oleh konsumen	
		13. Sistem akan menghitung total pembayaran secara otomatis
		14. Sistem akan menampilkan

		jika jumlah uang yang di bayar kurang akan tampil di bagian "total kekurangan" dan apabila jumlah yang dibayarkan lebih maka akan tampil di bagian "Total kembalian"
	15. Tekan tombol simpan untuk mengakhiri transaksi	
		16. Sistem akan menyimpan proses transaksi ke database
		17. Sistem akan menampilkan struk pembelanjaan
	18. Kasir akan mencetak stuk pembelanjaan	

3. Nama Use Case : *Membuat data barang*

   Aktor Utama : *admin*

   Aktor Tambahan : -

   Tujuan : menambah kategori barang,nama barang

Tabel 4.8. Skenario *Membuat data barang* yang di Usulkan

Admin	Sistem
1. Admin masuk kemenu master barang	
	2. Sistem akan menampilkan menu master barang
3. Admin memilih tambah data	
	4. Sistem akan menampilkan menu nama,katergori,ukuran,keterangan,harga produksi,harga jual,harga grosir,jumlah, warna, kode Gudang,nama Gudang
5. admin menginputkan nama,kategori,ukuran,keterangan,harga produksi,harga jual,harga grosir,jumlah,warana,kode supplier,gambar, kemudian setelah selesai tekan simpan	
	6. Sistem akan mengeluarkan pemberitahuan “berhasil”

4. Nama Use Case : *Mengelola laporan*

   Aktor Utama : *admin*

   Aktor Tambahan : -

   Tujuan : Melihat dan mencetak laporan barang dan laporan transaksi

Tabel 4.9. Skenario *Mengelola Laporan* yang di Usulkan

Admin	**Sistem**
1. Admin masuk menu laporan	
	2. Sistem akan menampilkan menu laporan produksi dan penjualan
3. Admin memilih pilihan menu	
	4. Sistem akan menampilkan menu tanggal dan jenis laporan (Laporan Penjualan)
5. Admin memasukan tanggal mulai masuk laporan sampai akhir laporan,jenis Laporan yang ingin ditampilkan (Laporan penjualan) Dan tekan tombol print	
	6. Sistem akan mencetak hasi laporan tersebut

5. Nama Use Case : *Mengelola Barang Masuk*

   Aktor Utama : *Gudang*

   Aktor Tambahan : *Supplier*

   Tujuan : menambah Barang yang Masuk

Tabel 4.10. Skenario *Mengelola barang Masuk* yang di Usulkan

Supplier	Gudang	Sistem
1. Supplier memberikan barang kepada bagian gudang		
	2. Bagian gudang menerima barang dari *supplier*	
	3. Bagian gudang menginputkan data barang ke sistem	
	4. Pilih menu pengelolaan barang masuk	
		5. Sistem akan menampilkan Menu pengelolaan barang masuk
	6. Tekan tombol tambah data	

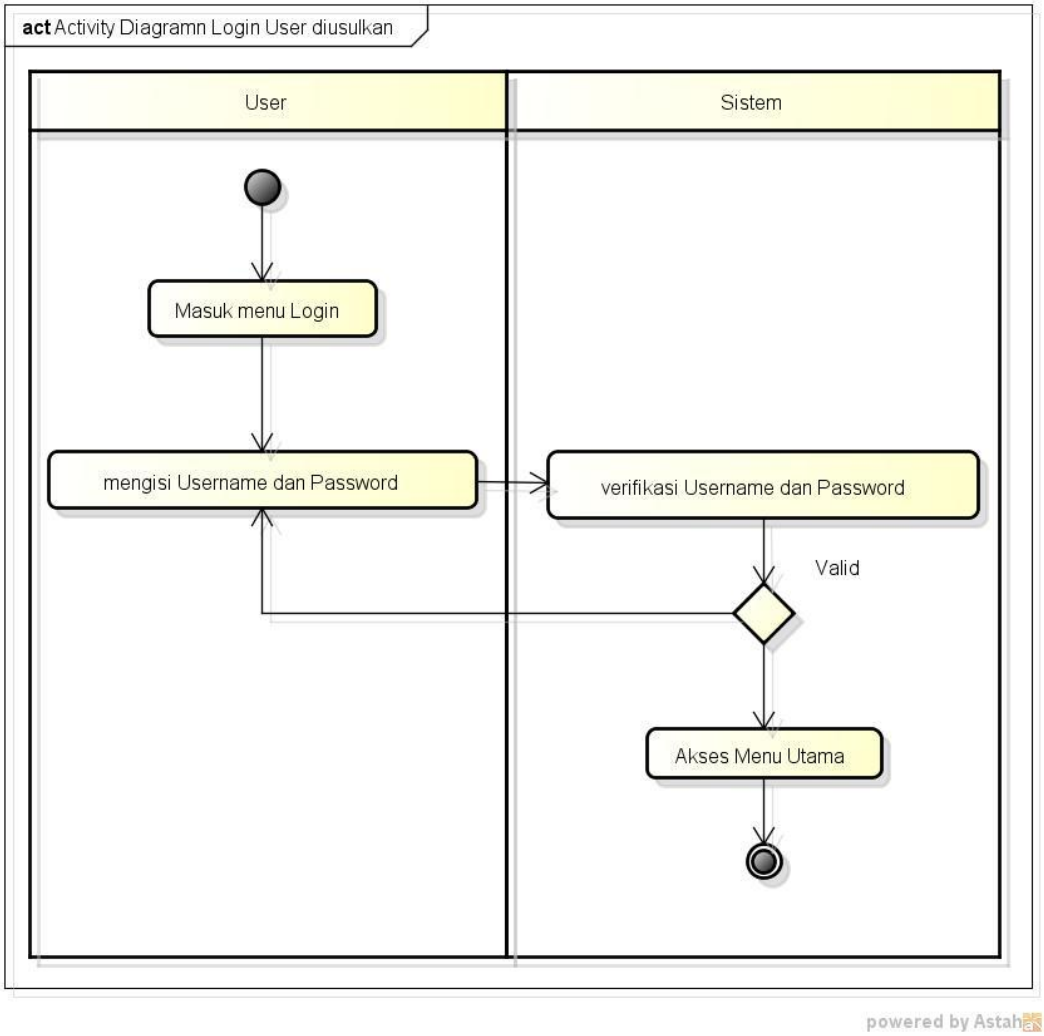
		7. Sistem Menampilkan menu Kategori,nama,warna,ukuran,
	8. Gudang menginputkan data kategori, nama, warna, ukuran, dan Jumlah	
		9. Sistem akan menampilkan Stok barang, harga produksi, kode barang, max stock, kode *supplier*, nama *supplier*
	10. Tekan tambah barang untuk dapat menambah jumlah barang yang dimasukan	
		11. Sistem akan menyimpan data barang yang telah diinputkann kedalam daftar barang
	12. Gudang menyimpan data barang kedata base ketika telah selesai menginpukan	
		13. Sistem akan menyimpan data tersebut.

4.2.2.3. Activity Diagram

Activity diagram berikut ini memperlihatkan secara rinci aliran data secara logika tanpa mempertimbangkan lingkungan fisik dimana data mengalir. User diharuskan untuk login terlebih dahulu jika ingin menggunakan dan menjalankan pelayanan-pelayanan yang diberikan.

1. Deskripsi Login User

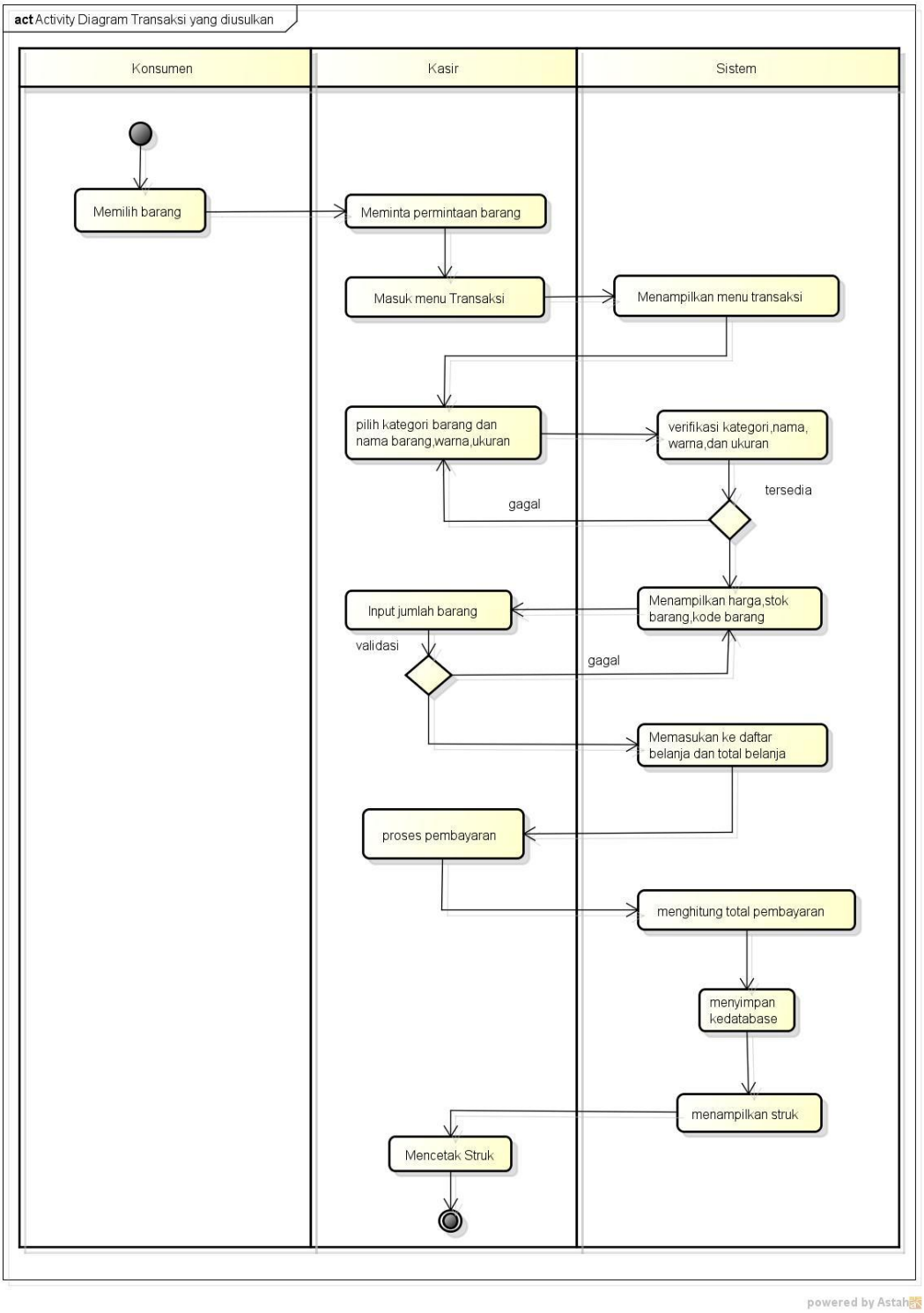
   User melakukan login, Jika *account* terdaftar, maka secara otomatis akan masuk ke halaman utama, jika tidak ada data memberikan peringatan 'Maaf Username dan Passworyang anda masukan salah' dan kembali ke halaman *login.*



Gambar 4.6. *Activity Diagram Login User* yang di Usulkan di Naura Fashion

2. Deskripsi proses Transaksi

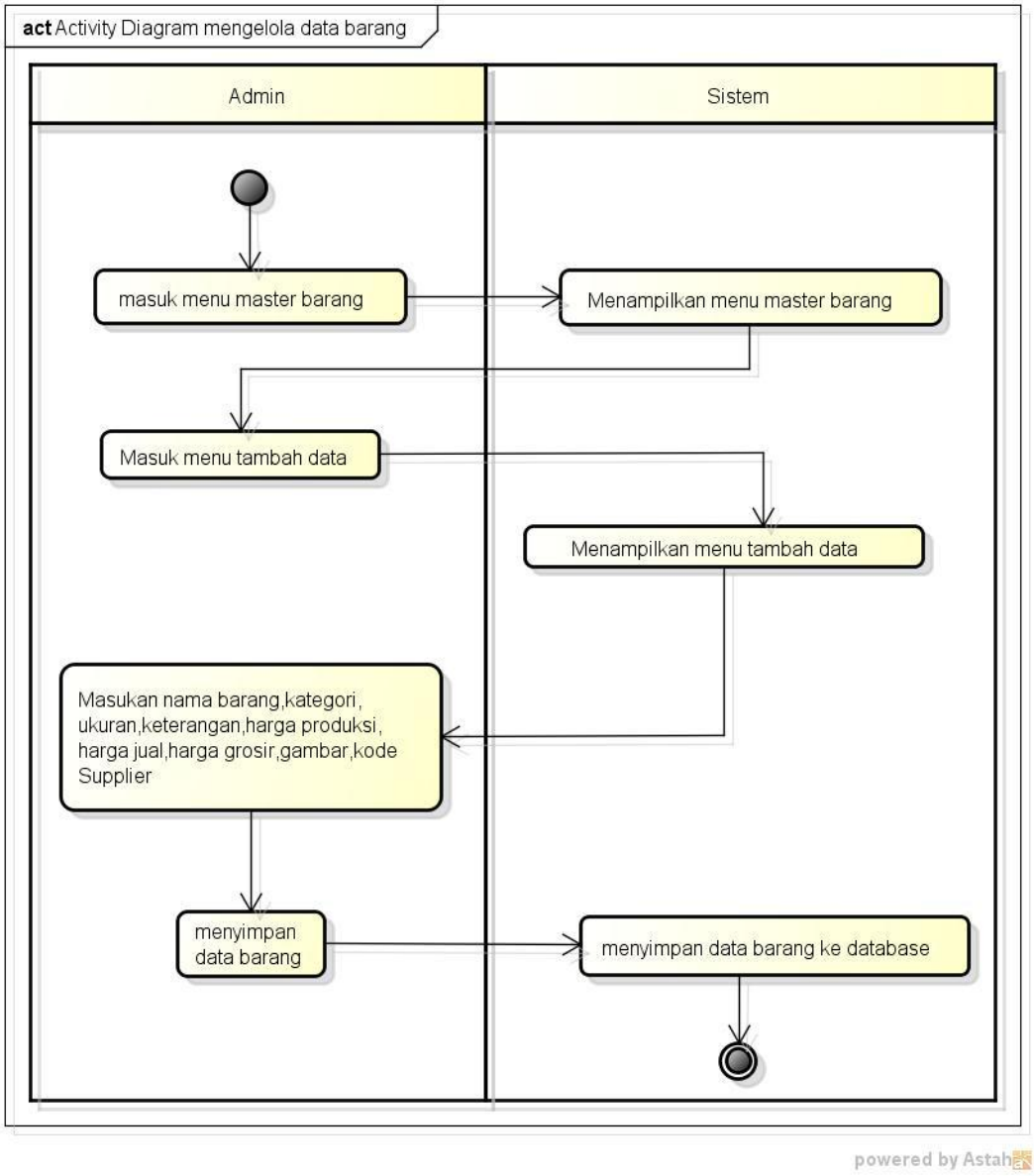
   Kasir Masuk Ke Menu Transaksi Barang lalu memasukan data-data yang diperlukan untuk transaksi, kasir hanya mengetikan nama barang dan jumlah barang yang di pesan kemudian melalui proses perhitungan otomatis, kasir tidak perlu lagi menghitung secara manual dan secara langsung data transaksi tersimpan di database.



Gambar 4.7. *Activity Diagram Proses Transaksi* yang di Usulkan di Naura Fashion

3. Deskripsi mengelola data barang

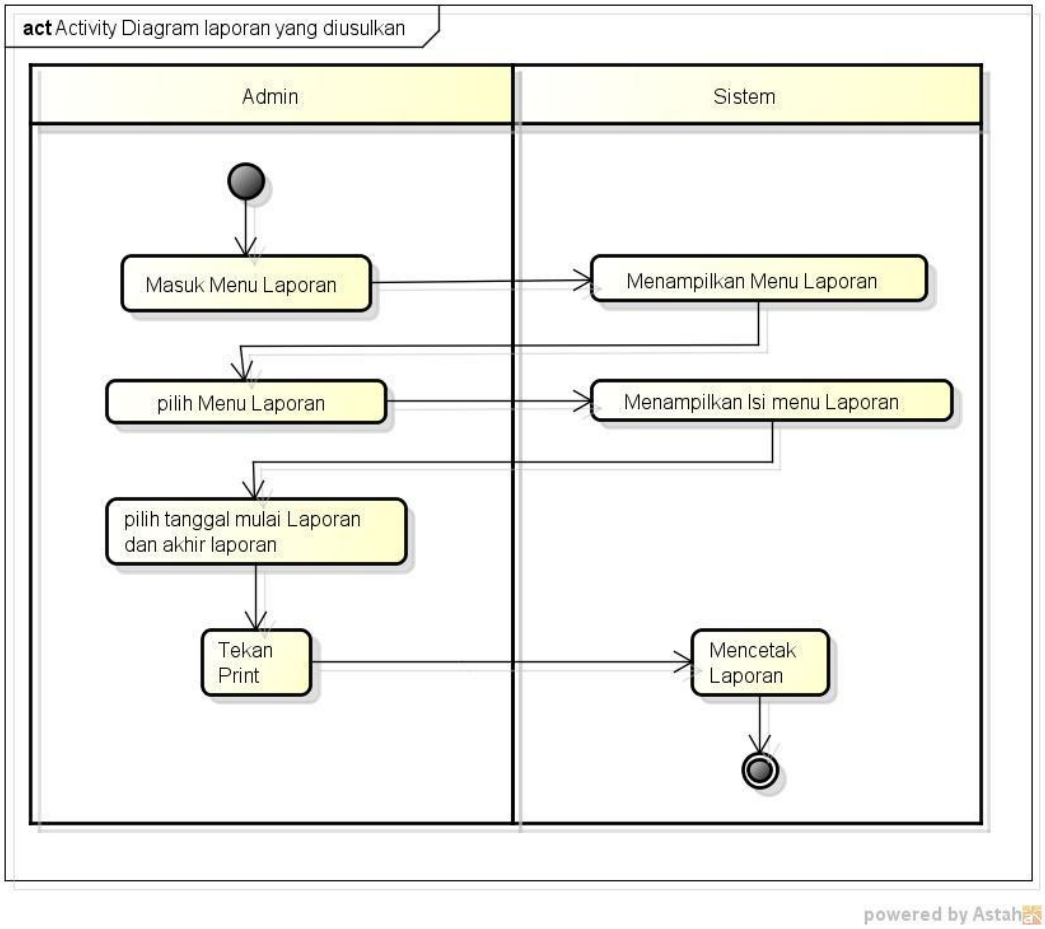
Admin dapat menambah nama barang,kategori barang dan stok awal barang,harga produksi,harga normal, harga grosir,dan nama Gudang yang akan tersimpan kedalam database.



Gambar 4.8. *Diagram Activity mengelola data barang* yang di usulkan di Naura Fashion

4. Deskripsi pengelolaan Laporan

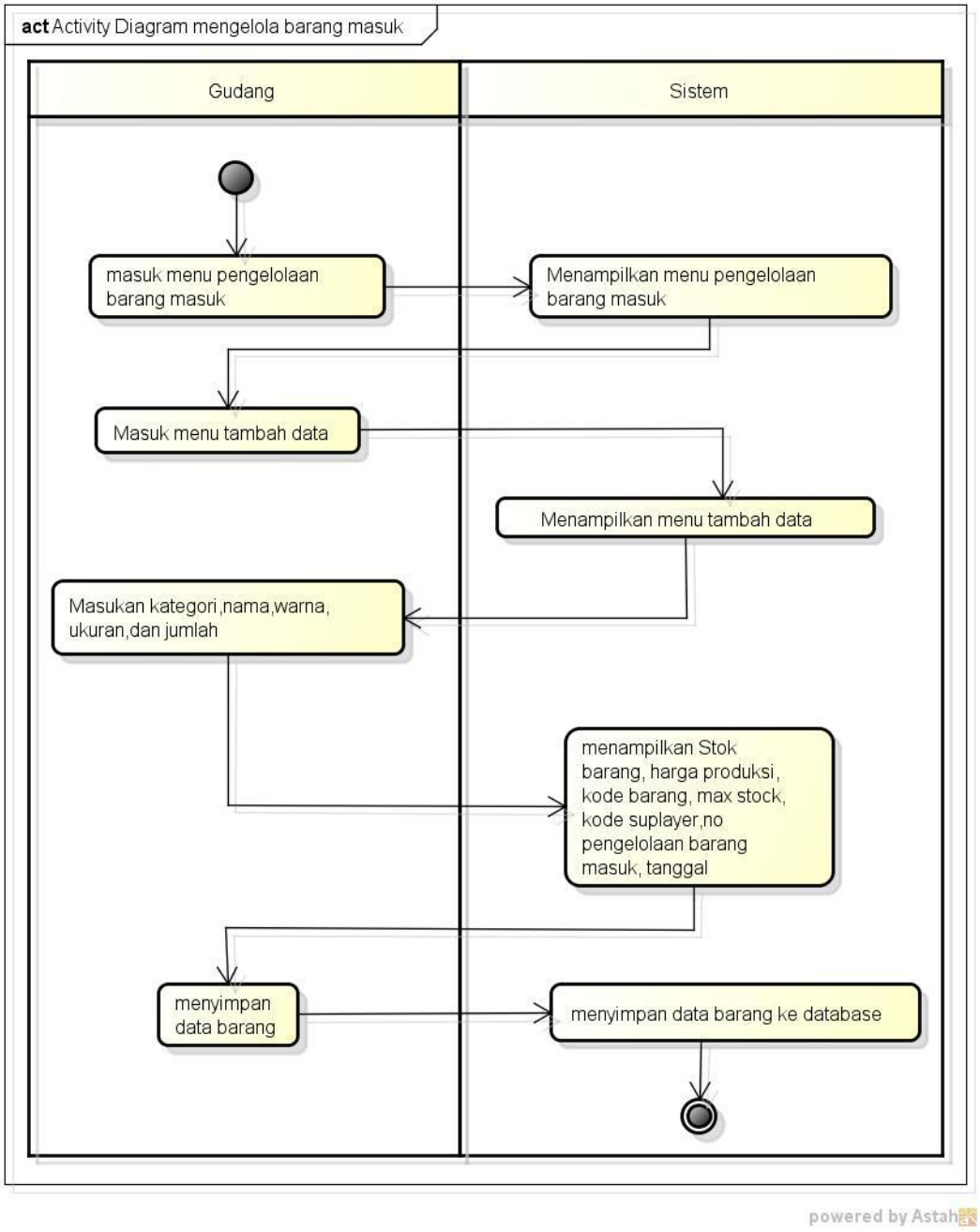
   Admin dapat melihat dan mencetak laporan barang masuk dan laporan penjualan setiap dibutuhkan yang sudah tersimpan di dalam database.



Gambar 4.9. *Diagram Activity mengelola Laporan* yang di usulkan di Naura Fashion

5. Deskripsi mengelola Barang masuk

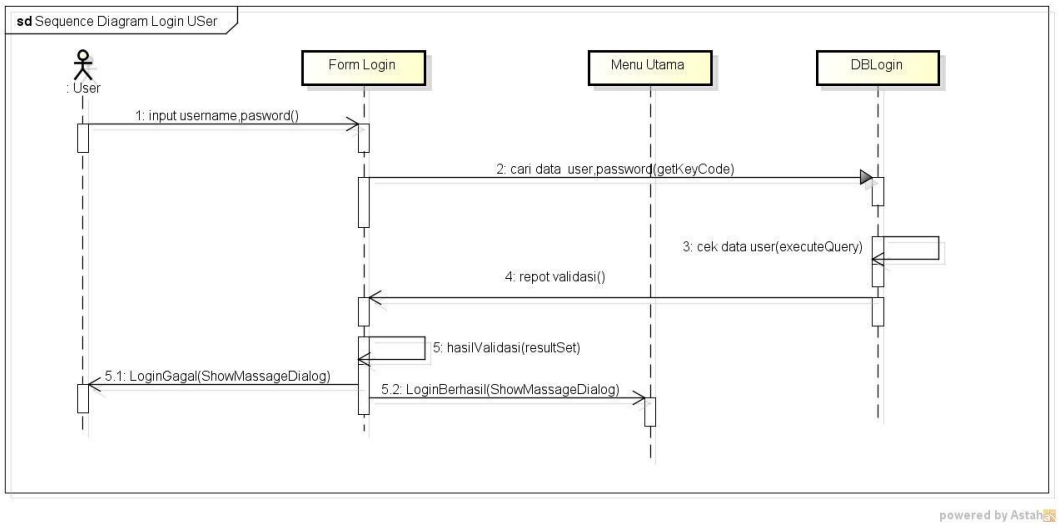
   Gudang dapat menambahkan jumlah barang yang telah tersedia dan datanya langsung tersimpan di dalam database dengan hanya menginputkan nama barang yang ada dan menambahakan jumlah barang yang telah tersedia.



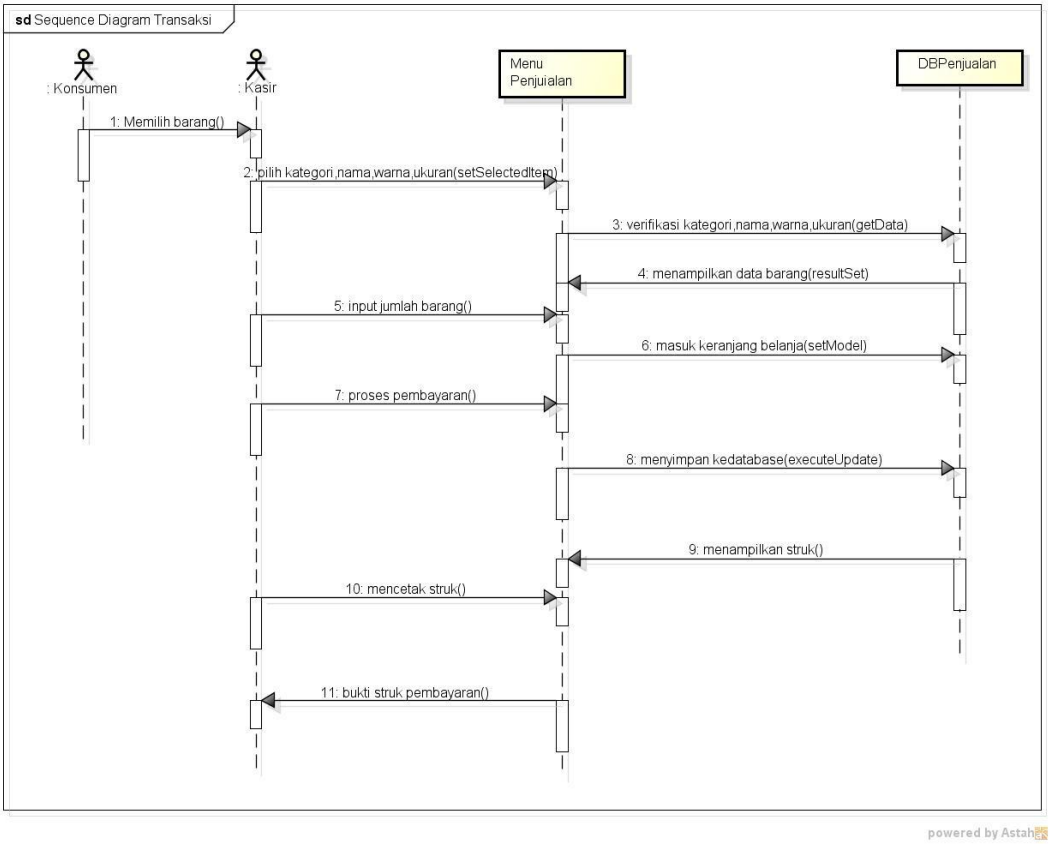
Gambar 4.10. *Diagram Activity mengelola Barang Masuk* yang di usulkan di Naura Fashion

4.2.2.4 Sequence Diagram

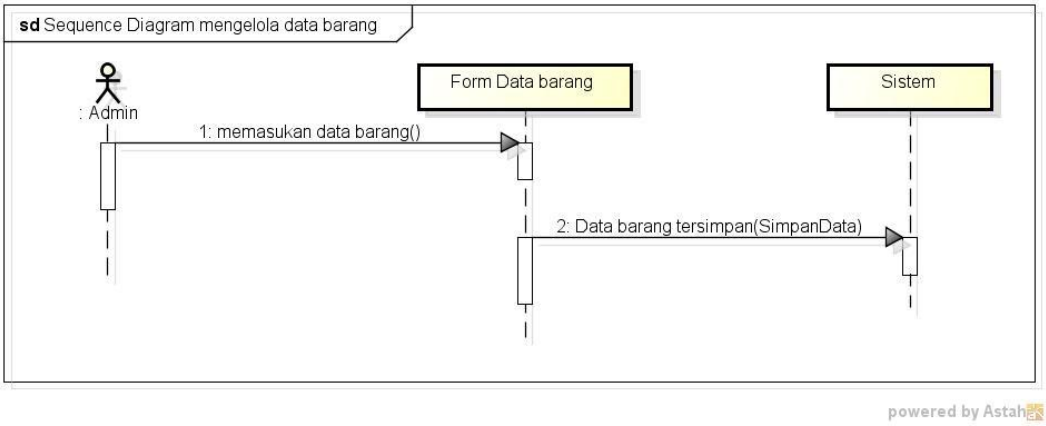
Menggambarkan kolaborasi dinamis antara sejumlah *object*. kegunaanya untuk menunjukan rangkaian pesan yang dikirim antara *object* juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.



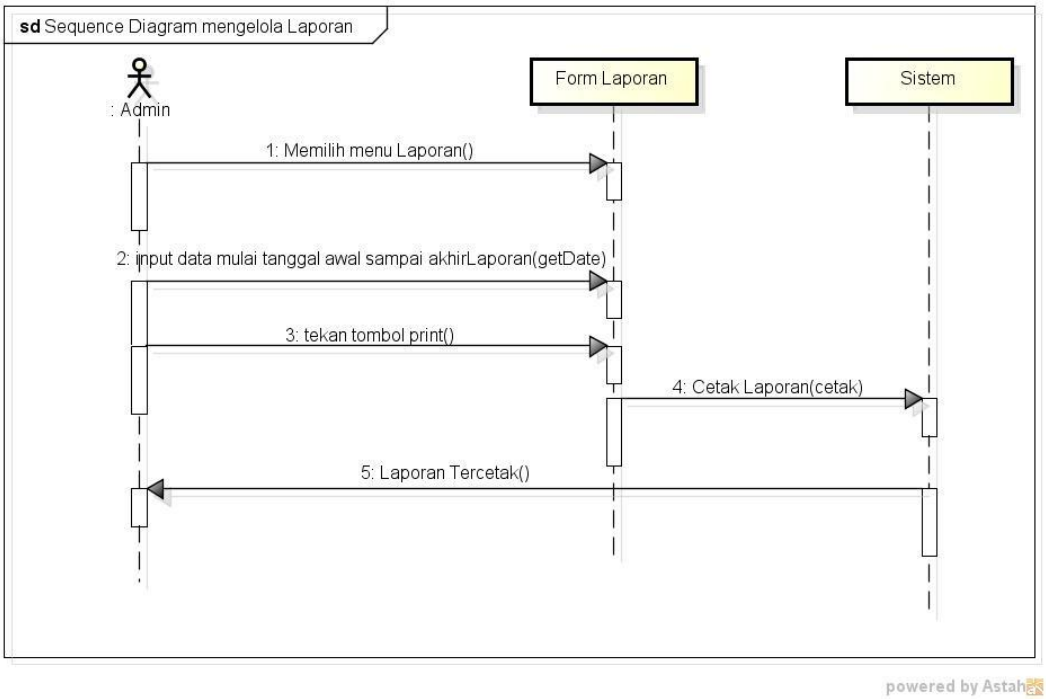
Gambar 4.11. *Sequence Diagram Login* Yang Diusulkan



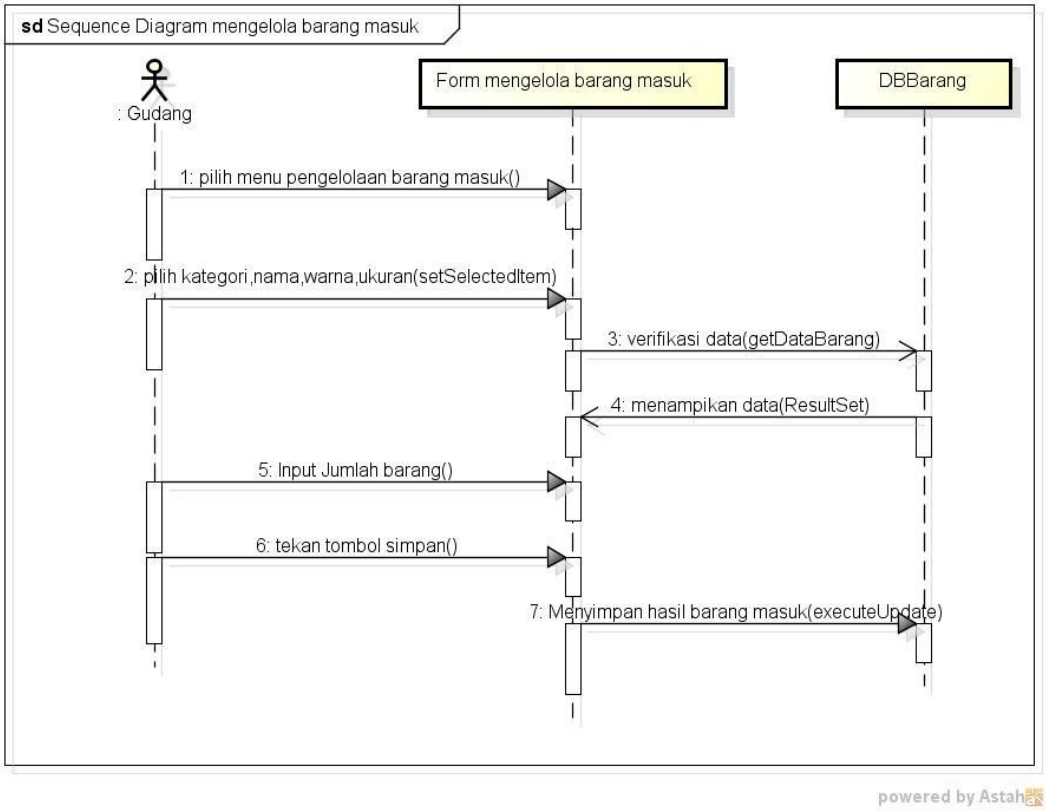
Gambar 4.12. *Sequence Diagram Transaksi* Yang Diusulkan



Gambar 4.13. *Sequence Diagram mengelola data barang* Yang Diusulkan



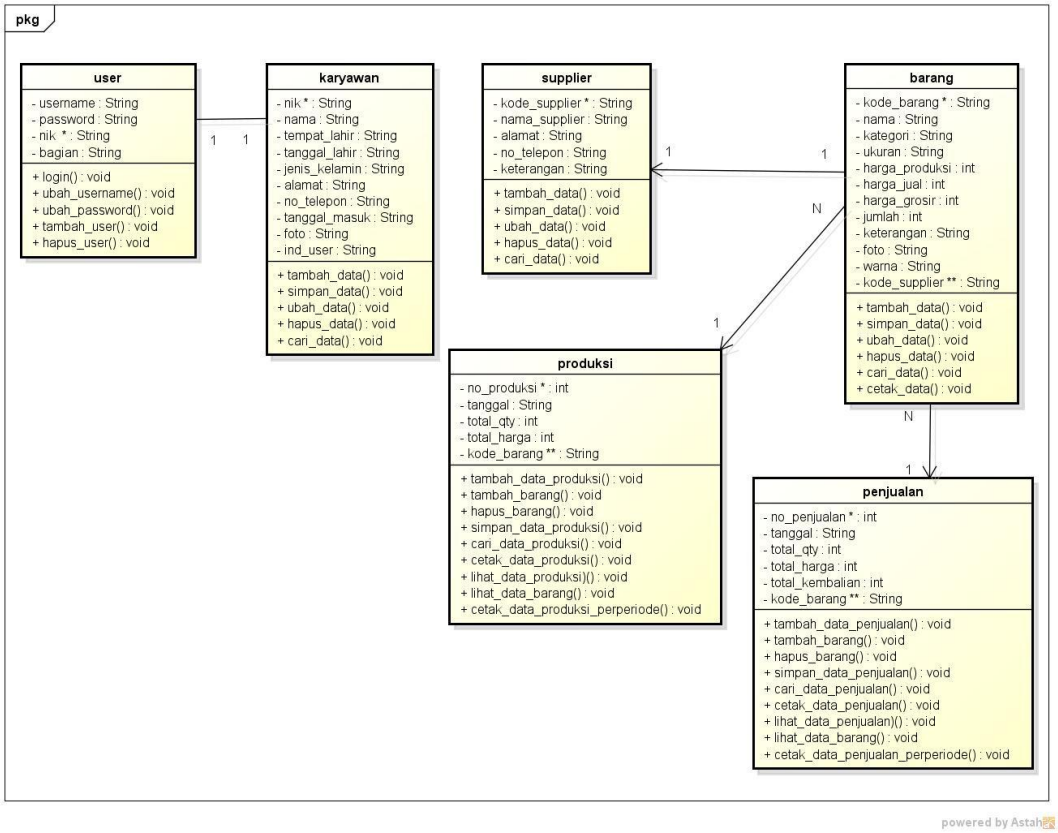
Gambar 4.14. *Sequence Diagram Mengelola Laporan* Yang Diusulkan



Gambar 4.15. *Sequence Diagram mengelola Barang Masuk* yang diusulkan

4.2.2.5. *Class Diagram*

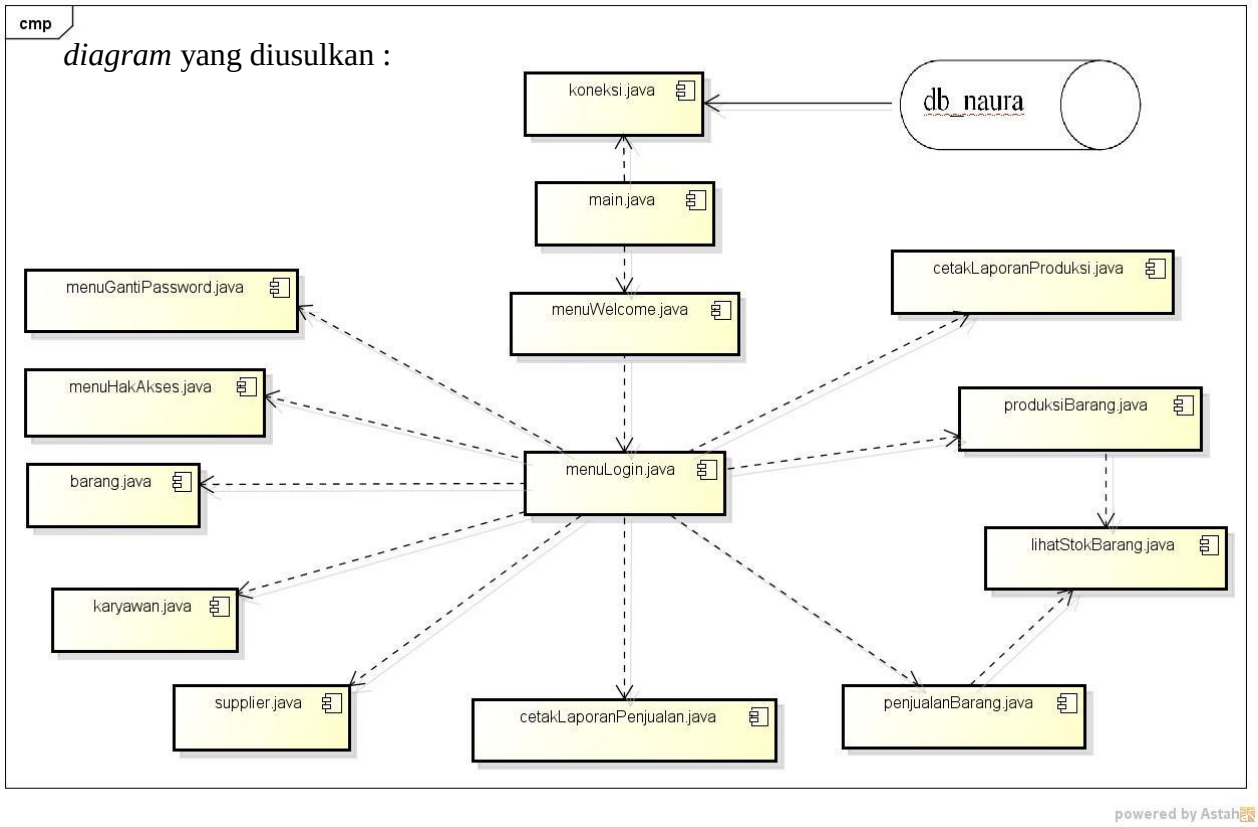
Class Diagram adalah sebuah spesifikasi yang jika di instansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class Diagram* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaam tersebut (metoda/fungsi). *Class Diagram* menggambarkan struktur dan deskripsi *Class*, *Package*, dan dan *Object* beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain. Berikut adalah *Class Diagram* yang dibutuhkan dalam perancangan Program Aplikasi pembantu dan pendukung kinerja karyawan di Naura Fashion :



Gambar 4.16. *Class Diagram* Yang Diusulkan

4.2.2.6. *Component Diagram*

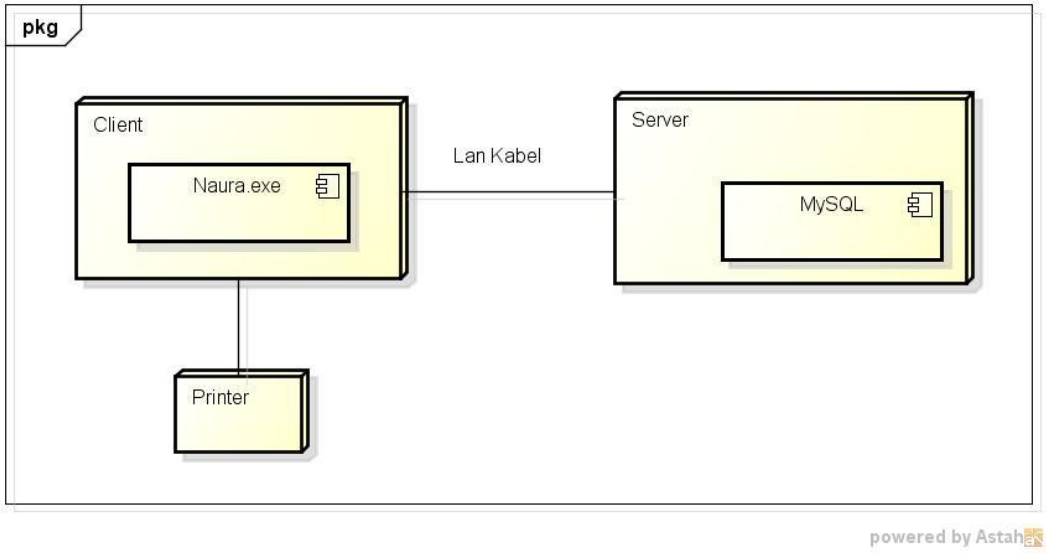
Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time*, maupun run *time*. Umumnya komponen terbentuk dari beberapa *class* dan/atau *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa *interface*, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain. Berikut gambar *Component diagram* yang diusulkan :



Gambar 4.17. *Component Diagram* Yang Diusulkan

4.2.2.3. *Deployment Diagram*

Diagram Deployment adalah diagram yang menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware dan menunjukan konfigurasi dalam proses eksekusi aplikasi.



Gambar 4.18. *Deployment Diagram* Yang Diusulkan

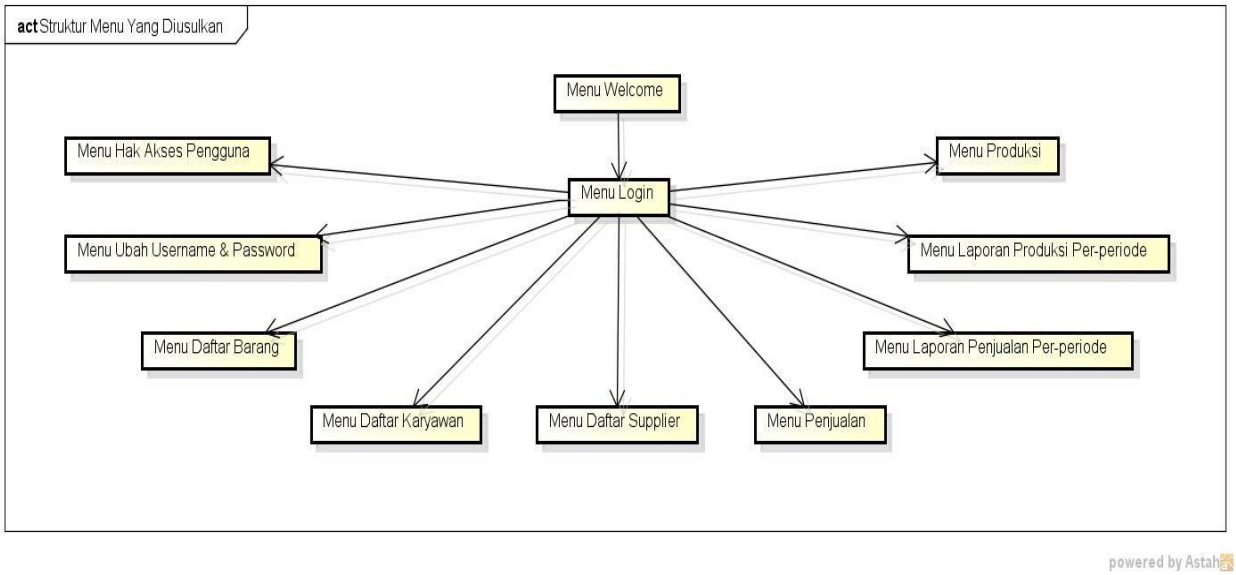
4.2.3. Perancangan Antar Muka

Pada perancangan antar muka atau *interface* ini akan menggambarkan rancangan tampilan yang ada dalam sistem yang diusulkan. Adapun yang akan dibahas adalah perancangan input dan perancangan *output*.

4.2.3.1. Struktur *Menu*

Struktur *menu* adalah bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer sehingga pada saat menjalankan program dikomputer, *user* tidak mengalami kesulitan lagi dalam memilih menu nyang diinginkan.

Struktur menu yang diusulkan adalah sebagai berikut :



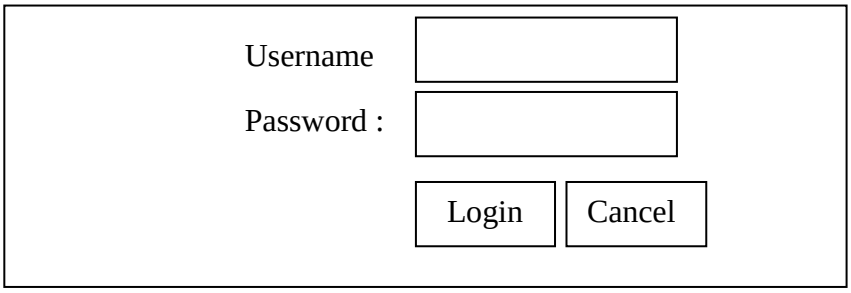
Gambar 4.19. Struktur *Menu* Yang Diusulkan

4.2.3.2. Perancangan *Input*

Perancangan ini dipergunakan bagi pengguna yang berkepentingan untuk menggunakan program aplikasi. Bagi yang menggunakan program aplikasi ini maka terlebih dahulu harus memasukkan *username* dan *password*.

1. *Login*

Tombol *Login* di *Form* Inputan *Login* akan menvalidasi langsung ke *database* sistem, dimana tombol *Login* diatas akan menampilkan pesan validasi *username* dan *password* dari *user* dan tomobo *Cancel* untuk mereset atau mengahapus ulang data yang sudah ditulis



Gambar 4.20. *Form* Inputan *Login* Yang Diusulkan

2. *Menu Ubah Username dan Password*

Tombol *Submit* di Form ubah Username dan Password adalah untuk memvalidasi langsung ke database dan masuk ke menu update Username dan Password merubah Username dan Pasword dimana tombil *Save* di Form Update Username dan Password adalah untuk menyimpan data Username dan Password yang baru ke dalam database.



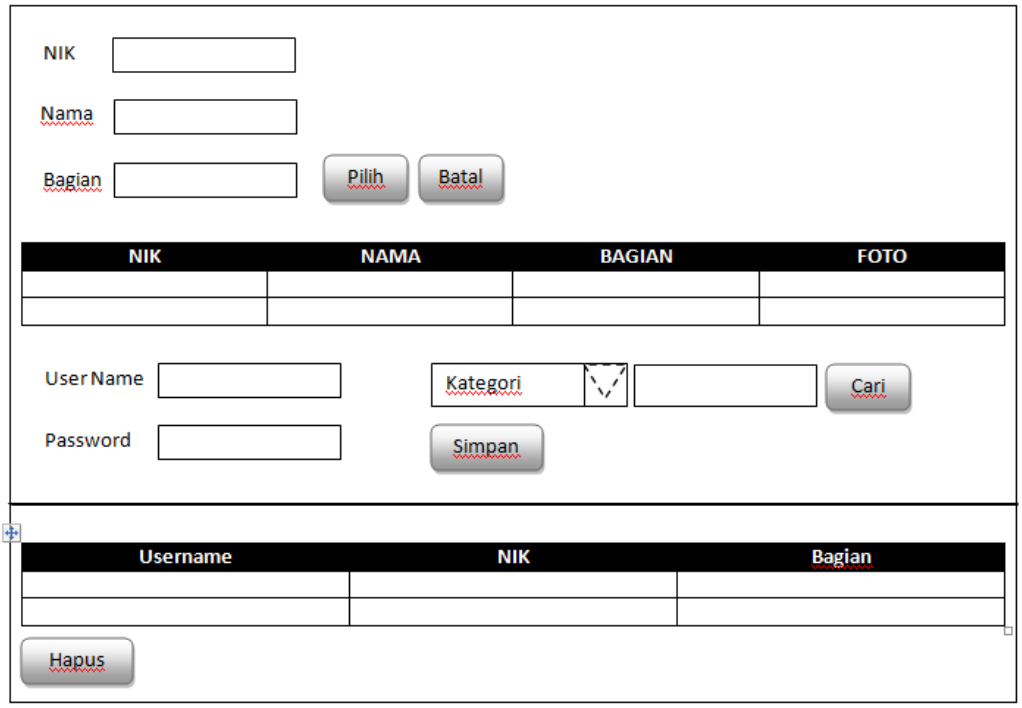
Gambar 4.21. *Form* Inputan *Ubah username dan password* Yang Diusulkan



Gambar 4.22. *Form* Inputan *Update username dan password* Yang Diusulkan

3. *Menu Hak Akses Pengguna*

Tombol *Pilih* dalam form Hak Akses adalah untuk memilih bagian pada database yang akan diberikan kepada user. Dan tombol *Simpan* adalah untuk menyimpan data user yang telah diberi hak akses oleh admin dan akan tersimpan kedalam database.

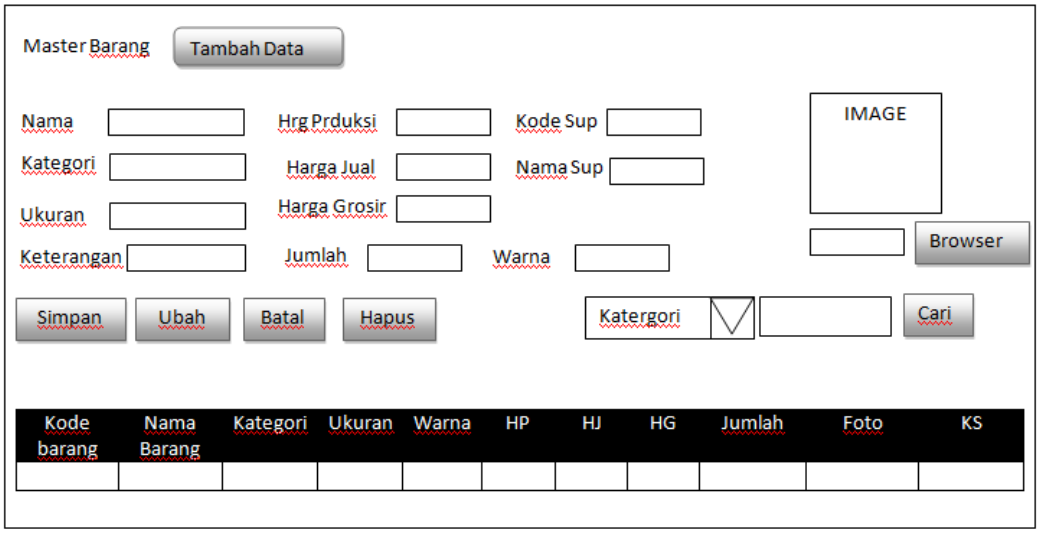


Gambar 4.23. *Form* Inputan *Hak Akses Pengguna* Yang Diusulkan

4. *Menu Master Barang*

Tombol *Tambah Data* didalam Form Master Barang untuk menginputkan data barang yang akan di simpan, tombol *Simpan* didalam Form Master barang adalah untuk menyimpan data barang yang sudah diinputkan,tombol *Ubah* adalah untuk merubah data yuyang sudah diinput,dan mengambil data didalam database, tombol *Hapus* adalah Untuk menghapus data barang yang telah disimpan di database. Disini terdapat tombol *Browser* dimana tombol

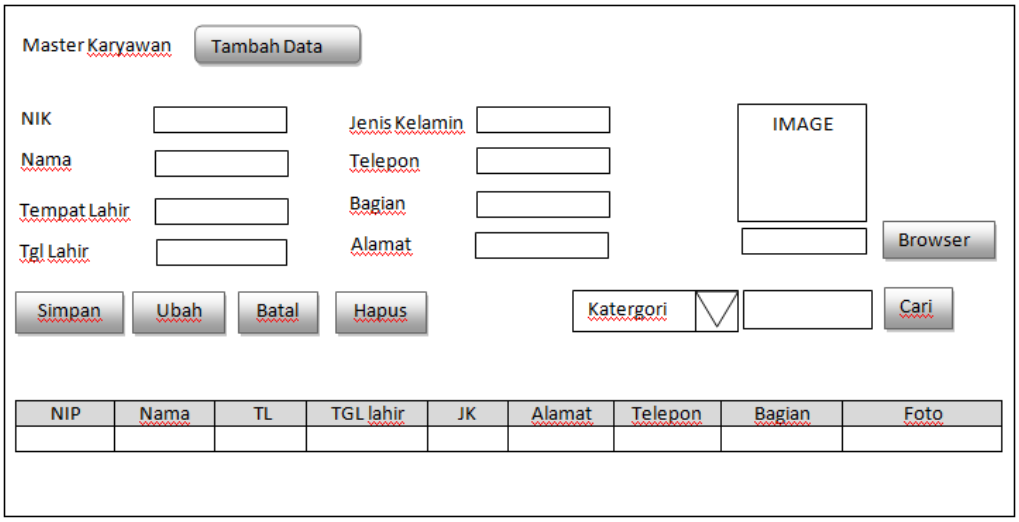
tersebut untuk memsukan Image/gambar dan Tombol *Cari* adalah untuk mencari data berdasarkan kategori yang dipilih.



Gambar 4.24. *Form* Inputan *Master Barang* Yang Diusulkan

5. Menu Master Karyawan

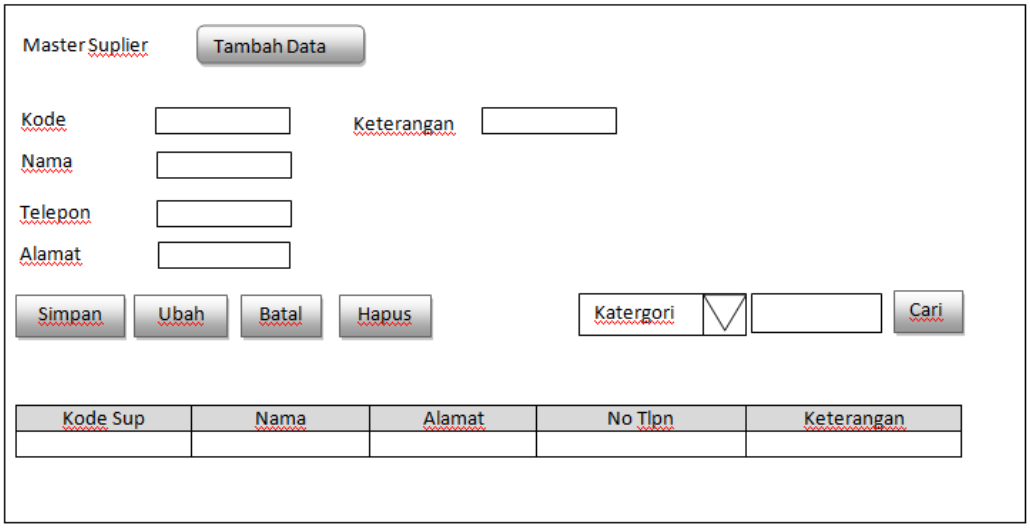
Tombol *Tambah Data* didalam Form Master Karyawan untuk menginputkan data barang yang akan di simpan, tombol *Simpan* didalam Form Master barang adalah untuk menyimpan data barang yang sudah diinputkan,tombol *Ubah* adalah untuk merubah data yuyang sudah diinput,dan mengambil data didalam database, tombol *Hapus* adalah Untuk menghapus data barang yang telah disimpan di database. Disini terdapat tombol *Browser* dimana tombol tersebut untuk memsukan Image/gambar dan Tombol *Cari* adalah untuk mencari data berdasarkan kategori yang dipilih.



Gambar 4.25. *Form* Inputan *Master Karyawan* Yang Diusulkan

6. Menu Master Supplier

Tombol *Tambah Data* didalam Form Master *Supplier* untuk menginputkan data barang yang akan di simpan, tombol *Simpan* didalam Form Master barang adalah untuk menyimpan data barang yang sudah diinputkan,tombol *Ubah* adalah untuk merubah data yuyang sudah diinput,dan mengambil data didalam database, tombol *Hapus* adalah Untuk menghapus data barang yang telah disimpan di database. Disini terdapat tombol *Browser* dimana tombol tersebut untuk memsukan Image/gambar dan Tombol *Cari* adalah untuk mencari data berdasarkan kategori yang dipilih.



Gambar 4.26. *Form* Inputan *Master Supplier* Yang Diusulkan

7. Menu Laporan

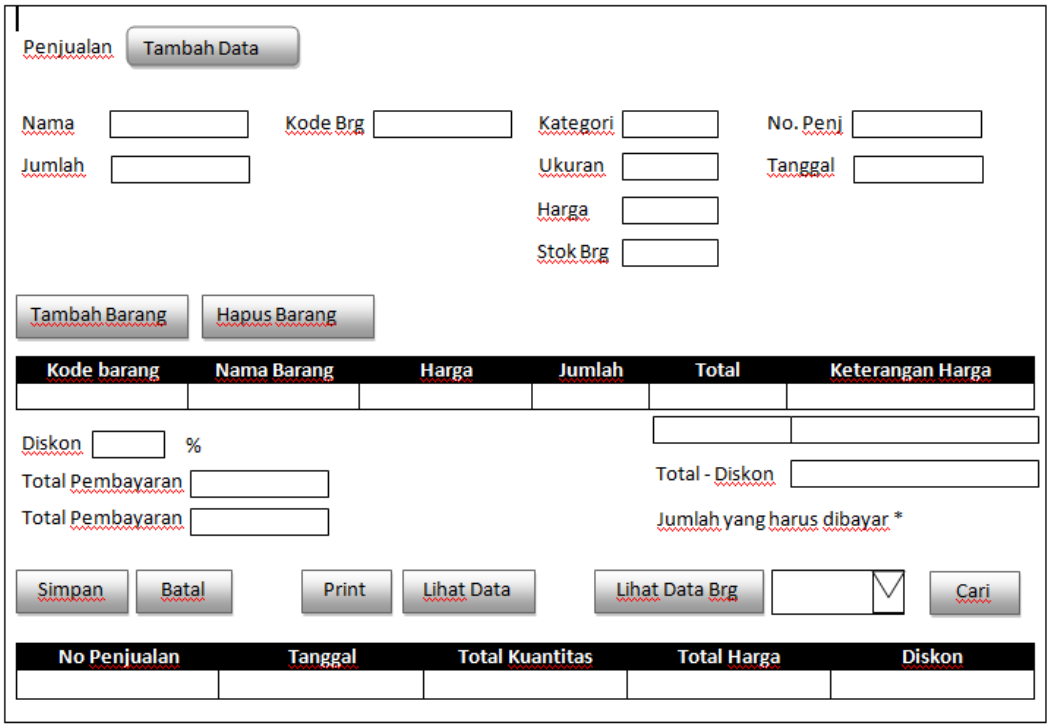
Dimenu ini terdapat tombol *Print* yang berfungsi untuk mencetak Laporan yang ada seperti Laporan Produksi dan Laporan Penjualan.



Gambar 4.27. *Form* Inputan *Laporan* Yang Diusulkan

8. Menu Transaksi

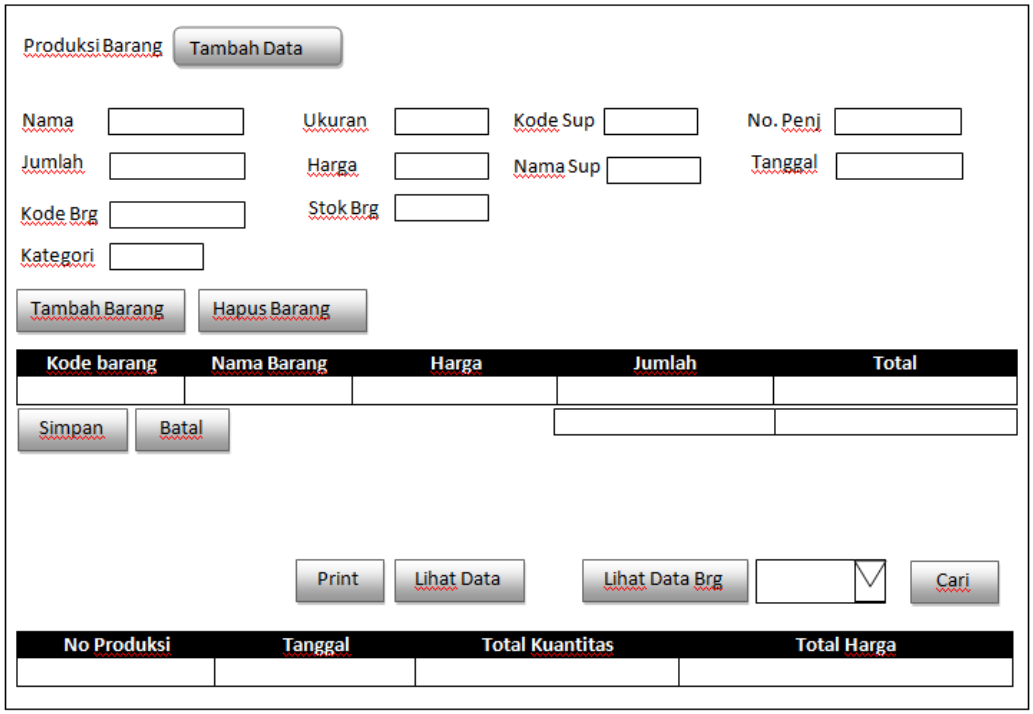
Tombol *Tambah Data* didalam Form Master Gudang untuk menginputkan data barang yang akan di simpan, tombol *Tambah Barang* adalah untuk menyimpan data barang yang sudah dipesan dan menambahkan barang atau jenis lain yang di pesan ,tombol *Hapus Barang* adalah untuk menghapus barang yang telah masuk keranjang belanja, tombol *Simpan* adalah untuk menyimpan hasil transaksi kedalam database, tombol *Batal* adalah untuk membatalkan semuah Transaksin yang bejalan, tombol *Print* adalah Untuk mencetak hasil transaksi yang berlangsung, tombol *Lihat data barang* yaitu tombol yang berfungsi untuk melihat semuah data barang yang tesedia. Dan tombol *Cari* adalah untuk mencari data transaksi yang dibutuhkan.



Gambar 4.28. *Form* Inputan *Transaksi* Yang Diusulkan

9. Menu barang masuk

Tombol *Tambah Data* didalam Form Master Gudang untuk menginputkan data barang yang akan di simpan, tombol *Tambah Barang* adalah untuk menyimpan data barang dan jumlah barang yang sudah diinputkan ,tombol *Hapus Barang* adalah untuk menghapus barang yang telah masuk keranjang belanja, tombol *Simpan* adalah untuk menyimpan hasil Produksi kedalam database, tombol *Batal* adalah untuk membatalkan semuah Produksi yang bejalan, tombol *Print* adalah Untuk mencetak hasil Produksi yang berlangsung, tombol *Lihat data barang* yaitu tombol yang berfungsi untuk melihat semuah data barang yang tesedia. Dan tombol *Cari* adalah untuk mencari data prduksi yang dibutuhkan.



Gambar 4.29. *Form* inputan mengelola barang masuk Yang Diusulkan

4.2.3.3. Perancangan *Output*

Rancangan *output* adalah produk dari sistem informasi yang dihasilkan setelah input dilaksanakan. Hasil dari rancangan input data terlihat dari media keluaran. Dalam rancangan *output*, sistem informasi ini juga memanfaatkan file database mengandung data-data yang terbilang cukup banyak sehingga membutuhkan tempat yang cukup luas untuk menggambarkan bagaimana sebenarnya rancangan *output* pada aplikasi yang akan dibuat.

1. Laporan Penjualan

NAURA FASHION

Jl. Waringin kelurahan Ciroyom Kec.Andir Bandung

Laporan Penjualan Per Tanggal : ……. sampai dengan ……..

Jenis Penjualan :

Tanggal	Nama Barang	Harga	Jumlah	Subtotal	Ket. Harga

Gambar 4.30. *Form* Output *Laporan Penjualan* Yang Diusulkan

2. Laporan Produksi

NAURA FASHION

Jl. Waringin kelurahan Ciroyom Kec.Andir Bandung

Laporan Produksi Per Tanggal : ……. sampai dengan ……..

Tanggal	Nama Barang	Nama Supplier	Harga	Jumlah	Subtotal

Gambar 4.31. *Form* Output *Laporan Produksi* Yang Diusulkan

3. Struk Transaksi

NAURA FASHION

Jl. Waringin kelurahan Ciroyom Kec.Andir Bandung

Penjualan :

No :

Tanggal :

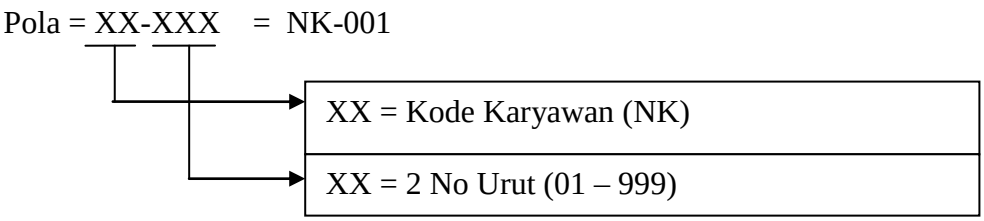
Nama	Kategori	Harga	Jumlah	Total

Gambar 4.32. *Form* Output *Struk* Transaksi Yang Diusulkan

4.2.3.4. Kodefikasi

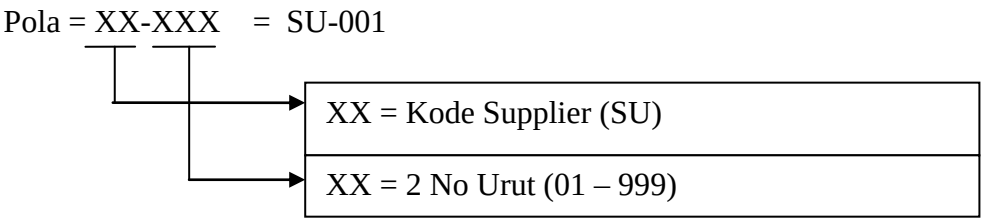
Merupakan pola kode otomatis yang dibentuk oleh sistem, dimana inputan yang dimasukkan akan otomatis digenerate oleh sistem. Contoh kedefikasi yang ada di Program Aplikasi ini adalah :

1. Kode Karyawan



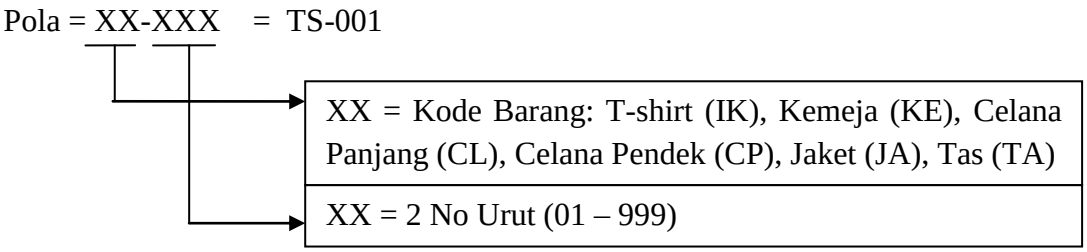
Gambar 4.33. Struktur Kodefikasi Kode Karyawan

2. Kode Supplier



Gambar 4.34. Struktur Kodefikasi Kode Supplier

3. Kode Barang

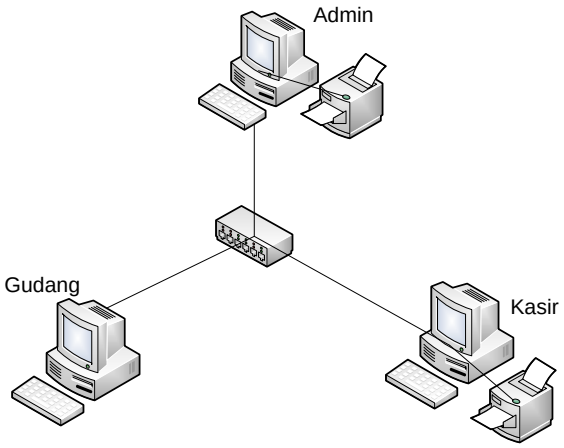


Gambar 4.35. Struktur Kodefikasi Kode Barang

4.2.3.5. Perancangan Arsitektur Jaringan

Topologi jaringan adalah bentuk hubungan dari suatu jaringan. Jenis topologi jaringan yang dipakai adalah *Point To Point* (titik ke titik).

Jaringan kerja titik ke titik merupakan jaringan kerja yang paling sederhana tetapi dapat digunakan secara luas. Begitu sederhananya jaringan ini, sehingga seringkali tidak dianggap sebagai suatu jaringan tetapi hanya merupakan komunikasi biasa. Dalam hal ini, kedua simpul mempunyai kedudukan yang setingkat, sehingga simpul manapun dapat memulai dan mengendalikan hubungan dalam jaringan tersebut. Data dikirim dari satu simpul langsung kesimpul lainnya sebagai penerima, misalnya antara terminal dengan CPU.



Gambar 4.36. Arsitektur Jaringan