



Studi Kasus

Perbaikan Pengelolaan Pergudangan Melalui Penerapan Sistem Informasi Pergudangan di CV. ABB

Hery Hamdi Azwir, Oktavia Patriani

Universitas Presiden, Kota Jababeka, Cikarang, Bekasi, 17550, Indonesia

INFORMASI ARTIKEL

Sejarah Artikel:

Diterima Redaksi: 8 Maret 2017

Revisi Akhir: 15 Mei 2017

Diterbitkan Online: 17 Mei 2017

KATA KUNCI

Sistem Informasi Manajemen Pergudangan
PHP7
Sistem Management Basis Data
MySQL
Basis data

KORESPONDENSI

Telepon: +62 21 89109762-63

E-mail: hery.azwir@president.ac.id

A B S T R A C T

CV ABB is a medium scale car rental company which manages around 150 vehicles consists of small car (sedan or MPV) until large bus. To supporting the maintenance of all vehicles, they build a warehouse for spareparts and tools. Unfortunately, the management of warehouse does not have a good governance because all records done manually. This situation creates very high risk for data manipulation because manual record method is done by warehouse officer, lost of spare part in warehouse, counterfeit receipt, and no system to control spare part in warehouse. An improvement need to be done by implementing a warehouse management information system to overcome this problem. By applying information system analysis and design concepts and implementing through the application of PHP7 and MySQL, the new system can control all spare part in warehouse including incoming and outgoing spare part, PIC who requests the spare part, checking stock of spare part in the warehouse and stock out used by finance to make purchase order, spare part report from supplier and spare part request report. This system is also able to help finance to create purchase order and get approval from the user in order to make the process is more transparent and accountable.

1. PENDAHULUAN

Dalam perindustrian diperlukan adanya manajemen pergudangan untuk mengontrol kegiatan pergudangan. Dengan adanya pengontrolan ini diharapkan dapat mengurangi biaya yang tidak perlu dikeluarkan untuk kebutuhan pergudangan, barang masuk dan barang keluar dari gudang harus benar dan sesuai dengan permintaan, keakuratan mengenai persediaan yang ada di gudang. Maka dari itu diperlukan adanya manajemen pergudangan [1,2].

CV. ABB bergerak dalam bidang jasa transportasi darat. Melayani antar jemput karyawan, sewa mobil kecil, mobil elf, bis $\frac{3}{4}$, bis besar, dan pariwisata. Gudang yang ada di CV. ABB digunakan untuk penyimpanan *spare part* dan alat yang digunakan untuk memperbaiki mobil yang digunakan untuk melayani pelanggan.

Manajemen Pergudangan di CV. ABB belum begitu baik seperti pendataan gudang, data *request part*, data *stock spare part*, data pengeluaran *spare part*, dan data *purchase order* yang masih sangat manual. Saat ini di CV. ABB belum memiliki alat bantu

pengawasan penggunaan *spare part* dan untuk pendataan *spare part* hanya di tulis tangan oleh karyawan, sehingga sangat rentan terjadinya kesalahan karyawan yang melakukan pencatatan dan kemungkinan manipulasi data. Sering terjadi kehilangan *spare part* di gudang, adanya bon palsu, dan belum adanya alat bantu pengawasan untuk mengawasi *spare part* yang ada di gudang. PHP dan MySQL adalah suatu tools open source dalam pengembangan sistem informasi berbasis web yang dapat dimanfaatkan untuk mengatasi permasalahan ini. Sistem informasi berbasis web ini akan membantu memudahkan pengawasan *spare part* karena murah dan relatif mudah untuk digunakan tanpa harus melakukan instalasi dari sisi client. Dalam mengatasi masalah yang ada perlu dilakukan perbaikan manajemen pergudangan agar manajemen gudang yang ada di CV. ABB bisa menjadi lebih baik, akuntabel dan transparan.

2. TINJAUAN PUSTAKA

2.1. Gudang

Menurut Yunarto dan Santika [3] terdapat beberapa tipe gudang seperti: *retail warehouse*, *central warehouse*, *manufacturing warehouse*, dan *distribution warehouse*. Gudang memiliki fungsi

sebagai tempat untuk menyimpan barang mentah, barang setengah jadi, dan barang yang siap untuk dipasarkan ataupun alat pekerja.

2.2. Manajemen Pergudangan

Manajemen pergudangan merupakan pengontrolan atau pengawasan kegiatan pergudangan yang diharapkan dapat mengurangi pengeluaran pembelian *spare part* atau keperluan gudang. Dalam manajemen pergudangan terdapat beberapa aktivitas yang terdiri atas kegiatan: administrasi, penerimaan barang, penyimpanan barang, pengepakan barang ke tempat yang dituju, dan pengeluaran barang.

Keuntungan yang didapatkan perusahaan dengan adanya pelaksanaan manajemen pergudangan yaitu: terhindar dari pengeluaran biaya yang tidak penting di dalam suatu perusahaan, dengan adanya manajemen pergudangan dapat meningkatkan perkembangan dan pengaturan bisnis di suatu perusahaan, mengantisipasi terjadinya manipulasi data dan kesalahan dalam input data khususnya *human error*.

2.3. Data Flow Diagram (DFD)

Menurut Hartono [4] DFD atau *data flow diagram* merupakan diagram yang menggambarkan arus data pada suatu sistem dengan menggunakan notasi simbol. DFD dapat juga menunjukkan hubungan antar komponen sistem. Simbol digunakan untuk memudahkan pembaca agar dapat melihat perbedaan antar proses, dan penyimpanan data.

Level 0 atau diagram *konteks* menggambarkan semua *entity* masuk dan keluar dari keseluruhan sistem, diagram ini adalah level yang paling tinggi di dalam DFD. Pada diagram level 0 tidak boleh ada komponen penyimpanan (*store*), yang ada hanya suatu proses. Ruang lingkup suatu sistem dapat digambarkan pada level ini. Diagram rinci merupakan uraian proses yang ada pada diagram suatu proses.

Setiap proses yang ada di DFD harus memiliki spesifikasi proses, untuk memudahkan pembaca dalam memahami DFD. Tabel 1 menunjukkan uraian spesifikasi proses pada DFD:

Tabel 1. Spesifikasi Proses

Keterangan	DeMarco and Yourdan Simbol	Gane and Sarson Simbol
Entitas Luar		
Proses		
Aliran data (data flow)		
Simpan data		

Sumber: [5]

Spesifikasi proses DeMarco dan Yourdan pada Tabel 1 proses memiliki simbol lingkaran, dan penyimpanan data dua garis panjang yang sejajar. Sedangkan spesifikasi proses Gane dan Sarson proses memiliki simbol persegi dan terdapat garis pemisah

pada atasnya dan untuk penyimpanan data memiliki simbol seperti persegi panjang namun pada sisi kanan tidak memiliki garis dan ada garis pemisah pada sudut kirinya. Nama proses pada DFD harus menunjukkan suatu pekerjaan atau suatu proses (perpaduan antara kata kerja dengan objek), contohnya membuat *purchase order*, menghitung persediaan gudang, mengirim *email*. Pada suatu proses harus ada entitas masuk dan entitas keluar yang berhubungan dengan proses, namun pada proses DFD tidak boleh ada nama proses yang sama.

2.4. Data Dictionary

Data dictionary merupakan sekumpulan data dan informasi fakta dari suatu sistem informasi sesuai dengan kebutuhan [6]. *Data dictionary* menyimpan nama pemakai, jenis - jenis akses yang diijinkan, data *item* dan nama *item* data dalam *database* untuk melakukan pemeriksaan tersebut. Sementara *data dictionary* dapat menyimpan ukuran dan jenis item data, batasan data *item*, dan nama *item* dalam pemeriksaan *integritas* data.

Data Dictionary terdiri dari berbagai macam yaitu: alias atau nama samaran, nama arus data yang mengalir pada DFD, arus data dari atau ke proses, penjelasan yang digunakan untuk menjelaskan keterangan - keterangan yang ada, bentuk data apakah dalam bentuk *file*, *print out* atau laporan, struktur data misal data arus terdiri dari apa saja yang tersimpan pada *data dictionary*, periode dan volume.

2.4.1 Simbol Data Dictionary

Simbol digunakan untuk membedakan satu nilai, satuan, *relevan* dan *elementer* data dengan nilai, satuan, *relevan* dan *elementer* data lainnya. Dalam melakukan pembedaan pemahaman dapat menggunakan notasi yang biasanya digunakan dalam menganalisis sistem menggunakan suatu kumpulan simbol:

Tabel 2. Simbol *data dictionary*

No.	Simbol	Uraian
1	=	Terdiri dari, mendefinisikan, diuraikan menjadi, artinya
2	+	Dan
3	()	Opsional (boleh ada atau boleh tidak ada)
4	{ }	Pengulangan
5	[]	Memilih salah satu dari sejumlah alternatif, seleksi
6	**	Komentar
7	@	Identifikasi atribut kunci
8		Pemisahan sejumlah alternatif pilihan antara simbol []

Sumber: [7]

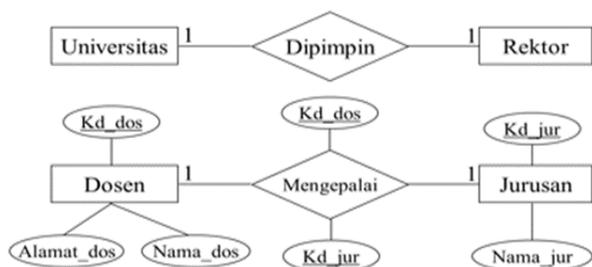
2.5. ERD

Entity Relationship Diagram atau disebut juga ER diagram digunakan dalam mendokumentasi kebutuhan pengguna dan dalam merancang *database* secara logis dengan menggunakan ER atau *semantic object* sesuai kebijakan perusahaan [8]. ER berisi tentang kumpulan entitas yang lengkap dengan sejumlah atribut dan sejumlah komponen relasi. Simbol tersebut terbagi atas dua simbol yaitu: atribut, *entity*, relasi, dan *link*. Atribut menguraikan tentang tujuan dari relasi atau *entity*. Entity atau entitas adalah sekumpulan objek nyata atau tidak nyata di mana data tersebut disimpan, biasanya suatu *entity* dinamakan dengan

kata benda. Relasi, biasa diberi nama dengan kata kerja dan merupakan hubungan yang terjalin antar sekumpulan *entity*. *Link*, merupakan penghubung antar *entity* dengan relasi dan atributnya.

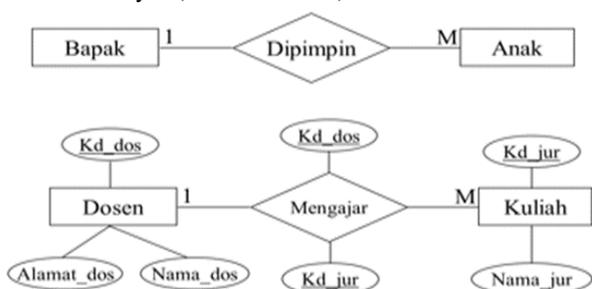
Dalam ERD terdapat tiga tingkatan hubungan yang terjadi yaitu:

a. Satu ke Satu



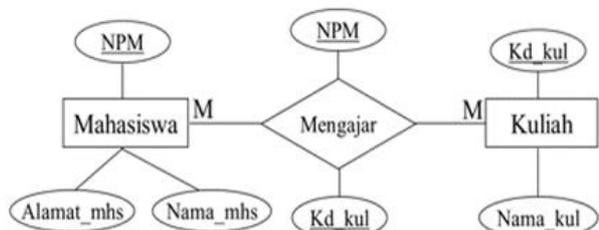
Gambar 1. ERD satu ke satu

b. Satu ke banyak (1 : M atau 1 : N)



Gambar 2. ERD satu ke banyak

c. Banyak ke banyak (N : N atau M : M)



Gambar 3. ERD banyak ke banyak

Dalam pembuatan ERD harus melalui beberapa tahap, berikut beberapa langkah - langkahnya:

1. Menentukan himpunan *entity* yang saling terlibat.
2. Menetapkan kunci atribut pada setiap himpunan *entity*.
3. Menentukan himpunan relasi pada himpunan *entity* dan *foreign key*nya.
4. Menetapkan kardinalitas pada masing - masing himpunan relasi.
5. Himpunan relasi dan *entity* beserta atributnya dilengkapi.

2.6. Database Design (Perancangan Database)

Perancangan *database* merupakan proses yang menghasilkan suatu model data rinci dari *database*, perancangan *database* melibatkan pemilihan: kolom untuk setiap tabel dan bagaimana tabel dan kolom berinteraksi satu sama lain [9]. Istilah perancangan *database* dapat berfungsi sebagai penerapan dari proses perancangan secara menyeluruh, tidak hanya pada struktur *databasenya*, tapi juga *query* dan bentuknya yang memiliki fungsi sebagai bagian dari *database* secara menyeluruh di dalam suatu sistem manajemen *database*.

Langkah - Langkah perancangan *database* yaitu:

1. Tentukan tujuan dibuatnya *database*.
2. Kumpulkan dan cari semua informasi yang dibutuhkan, misalnya nomor pesanan dan nama produk.
3. Bagi data yang sudah didapatkan ke dalam suatu tabel, lalu bagi setiap informasi dalam subjek atau *entity* besar.
4. Ubahlah setiap informasi ke dalam kolom dan putus informasi nama yang harus disimpan pada tabel.
5. Tentukan kunci primer, pilih *primary key* setiap tabel.
6. Aturlah relasi antar tabel.
7. Sempurnakanlah rancangan lalu analisis perancangannya.
8. Tetapkan aturan normalisasinya.

2.7. PHP7 Dan MySQL

Sejarah awal PHP adalah pendekatan dari *Personal Home Page* (situs personal), dulu PHP masih berbentuk *script* yang berfungsi sebagai pengolah data *form* dari *web server-side* yang memiliki sifat *open source*, dan memiliki nama *Form Interpreter* (FI) [10]. PHP (*Hypertext Preprocessor*) memiliki sintak yang mirip dengan ASP, *Java*, bahasa C, *Perl* dan memiliki kelebihan fungsi yang mudah dipahami dan spesifik. Kelebihan PHP yaitu:

- a. Bahasa pemrograman PHP tidak memerlukan kompilasi atau *Compiler* dalam penggunaannya.
- b. Memiliki sifat *open source*.
- c. Banyak aplikasi PHP gratis dan siap untuk digunakan seperti *PrestaShop*, *WordPress*, dll.
- d. Bisa membuat *web* menjadi dinamis.
- e. PHP memiliki banyak dukungan dari berbagai *web sever* contohnya saja *Apache*.
- f. Mudahnya mengembangkan aplikasi PHP
- g. PHP memiliki keunggulan lebih cepat dibandingkan dengan *Java* dan ASP.
- h. MySQL merupakan paket aplikasi dengan PHP.

Kekurangan PHP yaitu:

- a. PHP memiliki kelemahan keamanan.
- b. Kode PHP bisa dibaca oleh semua orang.
- c. Biaya untuk meng *encoding* membutuhkan biaya yang sangat mahal.

MySQL sebagai *database management system* dipakai secara bersamaan dengan aplikasi PHP agar aplikasi *server* lebih *powerfull* dan dinamis. Keuntungan MySQL yaitu:

- a. Portabilitas atau cukup stabil jika dijalankan.
- b. Lokalisasi atau bisa mendeteksi kesalahan pada klien.
- c. Konektivitas bisa menggunakan UNIX, IP, NT.
- d. MySQL bisa digunakan secara gratis.
- e. Keamanan terpercaya karena terdapat enkripsi kata sandi, izin akses *user*, dan nama *host*.
- f. Memiliki kecepatan yang tinggi dalam menangani *query* dan dapat memproses banyak SQL setiap detiknya.
- g. Lebih fleksibel dibandingkan dengan basis data yang lainnya.
- h. Dapat digunakan pada saat bersamaan tanpa adanya masalah oleh beberapa pengguna.
- i. *Interface* dengan semua aplikasi.

Kekurangan MySQL untuk Windows yaitu:

- a. Program hanya dapat *running* di *windows*.
- b. Aplikasi yang sedikit rentan terkena virus.

- c. Kurang *support* untuk koneksi kebahasa pemrograman *visual*.
- d. Tidak mampu menampung kinerja *server* yang melebihi batas maksimal.

3. METODOLOGI

3.1. Observasi Awal

Pada tahap awal dilakukan wawancara langsung terhadap *manager* dan *staff* di CV. ABB agar dapat mengetahui apa masalah yang sedang dihadapi oleh perusahaan. Observasi ini meliputi pengamatan langsung terhadap aktivitas perbaikan mobil, proses yang terjadi pada gudang yang berkaitan dengan penanganan suku cadang, dan keterkaitan dengan bagian keuangan. Pengamatan dilakukan baik saat jam kerja juga hari libur untuk mengetahui gambaran keadaan sebenarnya yang terjadi di lapangan. Hasil dari observasi ini adalah didapatnya analisis kebutuhan pengguna

3.2. Identifikasi Masalah

Hasil yang didapatkan dari observasi awal yang sudah dilakukan, selanjutnya merupakan landasan menentukan latar belakang masalah yang ada di CV. ABB, menentukan rumusan masalah, tujuan penelitian yang nantinya bisa menjawab semua masalah, dan batasan masalah agar penelitian tidak keluar dari ruang lingkup masalah.

3.3. Studi Literatur

Berikutnya dilakukan studi literatur untuk mendapatkan landasan teoritis maupun praktis untuk membantu pemecahan masalah yang dihadapi. Pada fase ini studi literatur dilakukan pada sejumlah referensi baik buku atau jurnal yang berkaitan dengan penelitian yang sedang dilakukan. Literatur yang digunakan adalah yang berkaitan dengan pengertian gudang, manajemen pergudangan, metodologi analisis dan perancangan sistem informasi yang diantaranya membahas DFD, *data dictionary*, *input/output design*, ERD, *database design*, dan literatur panduan untuk aplikasi PHP7 dan MySQL.

3.4. Pengumpulan Data

Pada tahap ini terdiri dari data primer dan data sekunder. Data yang dicari atau dibuat sendiri menurut pengamatan yang telah dilakukan disebut data primer. Sedangkan data sekunder yaitu data yang dapat dicari dari perusahaan. Data yang terkumpul dapat merupakan bukti atau *evident* adanya masalah yang menunjukkan bahwa solusinya adalah melalui perancangan sistem informasi.

3.5. Analisis dan Perancangan Sistem

Tahap analisis data yaitu menganalisis *user requirement*, *logical design*, *physical design*. *Logical design* meliputi analisis kebutuhan seperti DFD untuk mengetahui alur dan aliran data dari sistem informasi, dan *data dictionary* digunakan untuk menyimpan informasi, perancangan sistem meliputi ERD, *physical design*, *database design*, *input/output design*, *input/output design* digunakan untuk membuat tampilan layar dan formulir yang konsisten, ERD digunakan untuk mengetahui relasi antar *data master*, *view database design* dirancang untuk memenuhi kebutuhan informasi dari suatu perusahaan.

3.6. Kesimpulan dan Tindak Lanjut

Tahap kesimpulan menjelaskan perbedaan kondisi sebelum dan sesudahnya diterapkannya manajemen pergudangan. Dari hasil tersebut, diambil kesimpulan apakah dengan adanya manajemen pergudangan akan menjadi lebih baik atau tidak dan memberikan saran untuk tindak lanjut penelitian berikutnya.

4. HASIL DAN PEMBAHASAN

4.1. User Requirement

User requirement atau kebutuhan pengguna sistem informasi pergudangan di CV. ABB seperti *user*, gudang, *finance*, dan *admin* adalah sebagai berikut:

a. User

Kebutuhan *user* dalam sistem informasi untuk meminimalisir terjadinya kecurangan dibutuhkan info mengenai *stock spare part* yang ada di gudang agar *user* tidak memesan *spare part* yang tidak ada di gudang, untuk meminimalisir terjadinya kecurangan harus adanya bukti penerimaan *spare part* yang ditanda tangani oleh *user*.

b. Gudang

Kebutuhan gudang dalam pembuatan sistem informasi yaitu dibutuhkan informasi jika ada *spare part* yang sudah habis dan kurang dari 3 *spare part* agar gudang bisa mengingatkan *finance* untuk melakukan *order* kepada *supplier*, adanya *file purchase order* untuk mencocokkan *spare part* yang datang dengan *order* yang telah dibuat oleh *finance*, *print out* untuk *supplier* sebagai bukti penerimaan *spare part*, dan adanya informasi persediaan *spare part* yang ada di gudang tanpa harus melakukan pengecekan setiap *spare part*.

c. Finance

Finance memiliki kebutuhan dalam hal pengawasan *spare part* yang ada di gudang, maka dari itu dibutuhkan informasi *spare part* apa yang sudah habis dan mana saja yang masih ada di gudang, tampilan pengguna siapa saja yang dapat mengakses sistem informasi pergudangan, informasi mengenai *supplier* untuk memudahkan dalam pembuatan *purchase order*, adanya penyimpanan *purchase order* yang telah dibuat, *purchase order* yang telah dibuat dapat *diprint* untuk dilakukan persetujuan dari *admin*, dapat mengetahui informasi mengenai *spare part* yang telah dipesan apakah sudah sampai di gudang atau belum, adanya menu untuk melihat *user* yang melakukan pemesanan *spare part* ke gudang, dapat melakukan pengawasan *spare part* seperti *stock spare part* yang ada di gudang, dan *stock spare part* yang sudah habis. *Report* yang dibutuhkan dalam *finance* yaitu *report stock spare part* yang sudah habis, *report stock* yang ada di gudang, *report receive item*, dan *report users*.

d. Admin

Kebutuhan *admin* yaitu bisa melakukan pengawasan terhadap *user*, gudang, dan *finance* seperti melihat dokumen atau *report* yang ada pada setiap pengguna, bisa melakukan pengeditan dan modifikasi *file* pada setiap tampilan.

4.2. Analysis of Current System

Pada *analysis of current system* membahas tentang identifikasi titik keputusan, analisis proses bisnis yang mencakup dokumen flow diagram, DFD (*logical diagram*), dan identifikasi titik kelemahan.

4.2.1. Identifikasi Titik Keputusan

Berdasarkan dari *user requirement* didapatkan titik keputusan untuk melakukan perbaikan yaitu:

a. User

Tampilan menu user pada sistem informasi yang akan dibuat yaitu adanya menu *request part* yang digunakan untuk melakukan pemesanan *spare part*, menu *stock spare part* untuk melakukan pengecekan *stock spare part* yang ada di gudang, menu *change password* agar *user* bisa merubah sendiri *password* sesuai keinginan *user*, adanya tampilan *request part list* agar *user* bisa melihat kembali pesanan yang sudah di *request* jika ada kesalahan *request part user* bisa langsung mengkonfirmasi kepada *admin* untuk membatalkan pesanan atau mengubah pesanan sesuai dengan kebutuhan.

b. Gudang

Gudang merupakan tempat untuk menyimpan *spare part* sebelum digunakan oleh *user* untuk memperbaiki mobil. Maka dari itu pada tampilan menu gudang ditambahkan peringatan jika ada *spare part* yang sudah habis dan kurang dari 3 *spare part* agar gudang bisa mengingatkan *finance* untuk melakukan *order* kepada *supplier*, adanya *part master* untuk mengetahui *spare part* apa saja yang ada di gudang, menu *part type master* untuk menambahkan *part type* yang ada di gudang, *change password* untuk memudahkan dalam mengingat *password*, menu *purchase order list* untuk mencocokkan antara *spare part* yang datang dari *supplier* dengan *purchase order* yang telah dibuat oleh *finance*, menu *good receive* untuk memberi informasi bahwa *spare part* yang diorder sudah datang, menu *print out* pada *good receive* sebagai bukti kepada *supplier* bahwa *spare part* yang diorder sudah diterima, menu *request list* untuk melihat apakah ada *request part* dari *user* atau tidak dan melakukan *posting* jika *request part* sudah disetujui oleh *finance*, dan menu *stock master* untuk mengetahui persediaan *spare part* yang ada di gudang.

c. Finance

Pada tampilan menu *finance* terdapat menu peringatan untuk mengetahui *spare part* apa yang sudah habis dan mana saja yang masih ada di gudang, menu *part master* untuk mengetahui *part* apa saja yang ada di gudang, *user master* untuk mengetahui siapa saja yang dapat mengakses sistem informasi pergudangan, *part type master* untuk melakukan penambahan *part type* jika tidak ada dalam pilihan *part type*, menu *supplier master* untuk melakukan penambahan atau pengeditan *supplier* yang bekerjasama dengan perusahaan, *change password*, menu *purchase order list* untuk mengetahui *order* yang telah dibuat sebelumnya dan untuk membuat *purchase order* yang akan diberikan kepada *supplier* dalam melakukan *order spare part*, menu *good receive* untuk mengetahui apakah *spare part* yang diorder sudah datang atau belum, menu *request list* untuk melakukan *reject* atau *approve request part*, menu *stock master* untuk mengetahui *spare part* apa saja yang masih ada di gudang, menu *vehicle master* untuk melakukan penambahan mobil jika terdapat mobil baru dan untuk melihat rinci mobil yang ada di perusahaan, dan harus terdapat *report* untuk melakukan pengawasan *spare part* yang ada di gudang seperti *request user* untuk mengetahui siapa saja *user* yang sudah menerima *spare part*, *receive item* untuk mengetahui penambahan *spare part* dari *supplier*, *stock out* untuk mengetahui *spare part* apa saja yang sudah tidak ada di gudang.

d. Admin

Pada menu admin harus lengkap dari mulai dari tampilan menu *user*, gudang dan *finance* harus ada karena *admin* merupakan posisi tertinggi pada sistem informasi. Menu tambahan untuk admin diluar menu yang ada pada tampilan *user*, gudang dan *finance* adalah menu *action* yaitu *edit*, *add* atau *modification file* yang tidak boleh ada pada kebutuhan yang lainnya, seperti menu *location* untuk menambahkan dan mengedit lokasi *spare part* disimpan. *Admin* juga bisa melakukan *edit password user* yang sudah tidak dibolehkan membuka sistem informasi karena sudah keluar perusahaan.

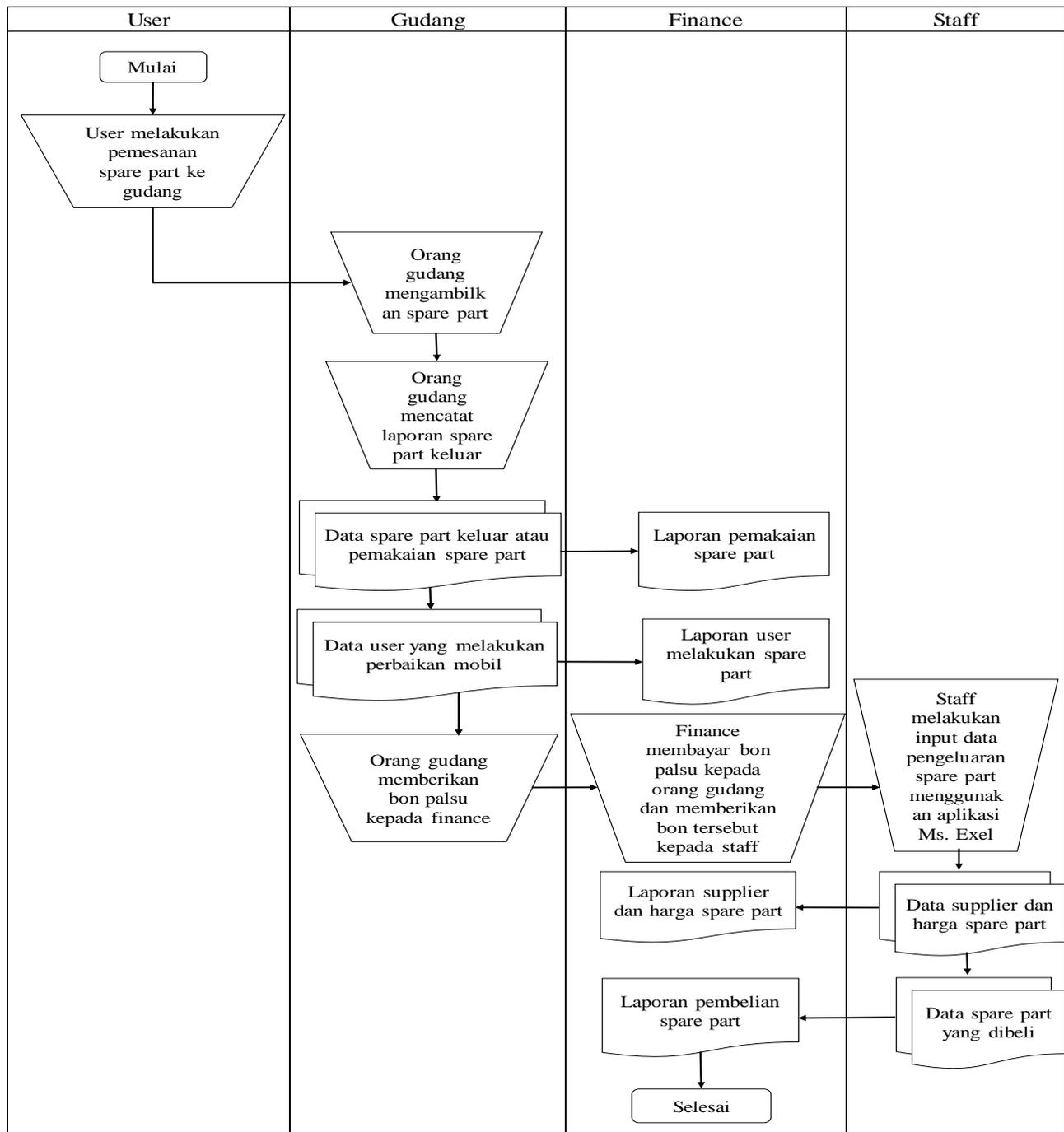
4.2.2. Analisis Proses Bisnis

Pada Gambar 4 menunjukkan alur manajemen pergudangan sebelum adanya sistem informasi di CV. ABB yaitu sebagai berikut:

- User* datang ke gudang untuk melakukan pemesanan *spare part*.
- Orang gudang mengambilkan *spare part* yang dipesan. Setelah orang gudang memberikan *spare part* kepada *user*, maka orang gudang mencatat *spare part* keluar secara manual pada buku atau kertas lembaran.
- Data *spare part* keluar atau pemakaian *spare part* dan data *user* yang melakukan perbaikan dibuat laporan pemakaian *spare part* dan laporan *user* melakukan *spare part*.
- Orang gudang menukarkan bon palsu kepada *finance*, setelah *finance* menerima bon palsu, *finance* membayar bon palsu tersebut kepada orang gudang dan memberikan bon tersebut kepada *staff* untuk diinput ke dalam data *supplier*, harga *spare part* dan data *spare part* yang dibeli, sehingga menjadi laporan *supplier*, harga *spare part* dan laporan pembelian *spare part*.

Proses bisnis awal yang dijelaskan pada Gambar 4 selanjutnya diperbaiki sehingga dapat dikembangkan menjadi suatu sistem informasi pergudangan yang akuntabel dan transparan yang diperlihatkan dalam Gambar 5. Pada Gambar 5 dapat dilihat bahwa alur dokumen dari sistem informasi yaitu pertama *log in* agar bisa masuk ke dalam sistem informasi mengisi *user name* dan *password* yang sesuai, kedua untuk *user* melakukan *request part* sesuai dengan kebutuhan mobil yang akan diperbaiki, setelah melakukan *request part* maka akan muncul *request user* pada *database request part master*.

Finance memiliki tugas untuk melakukan *reject* jika *part* rusak atau *user* tidak diijinkan mengambil *spare part* dan *approve* jika *user* diijinkan mengambil *spare part*, jika di *approve* maka gudang bisa mengambilkan *spare part* untuk *user* dan menyerahkan *spare part* kepada *user*, lalu pada sistem informasi, gudang harus melakukan *posting request part* agar *stock spare part* otomatis berkurang sesuai *request part*. Untuk memberikan bukti bahwa *spare part* sudah diterima oleh *user* maka gudang melakukan *print out proof of goods receipt* dan *user* harus menandatangani. Pada laporan data pergudangan berisi data *request user*, data *receive item*, data *stock out*, data *stock* yang kemudian pada menu *report* berisi *report request user*, *report receive item*, *report stock out*, dan *report stock*.



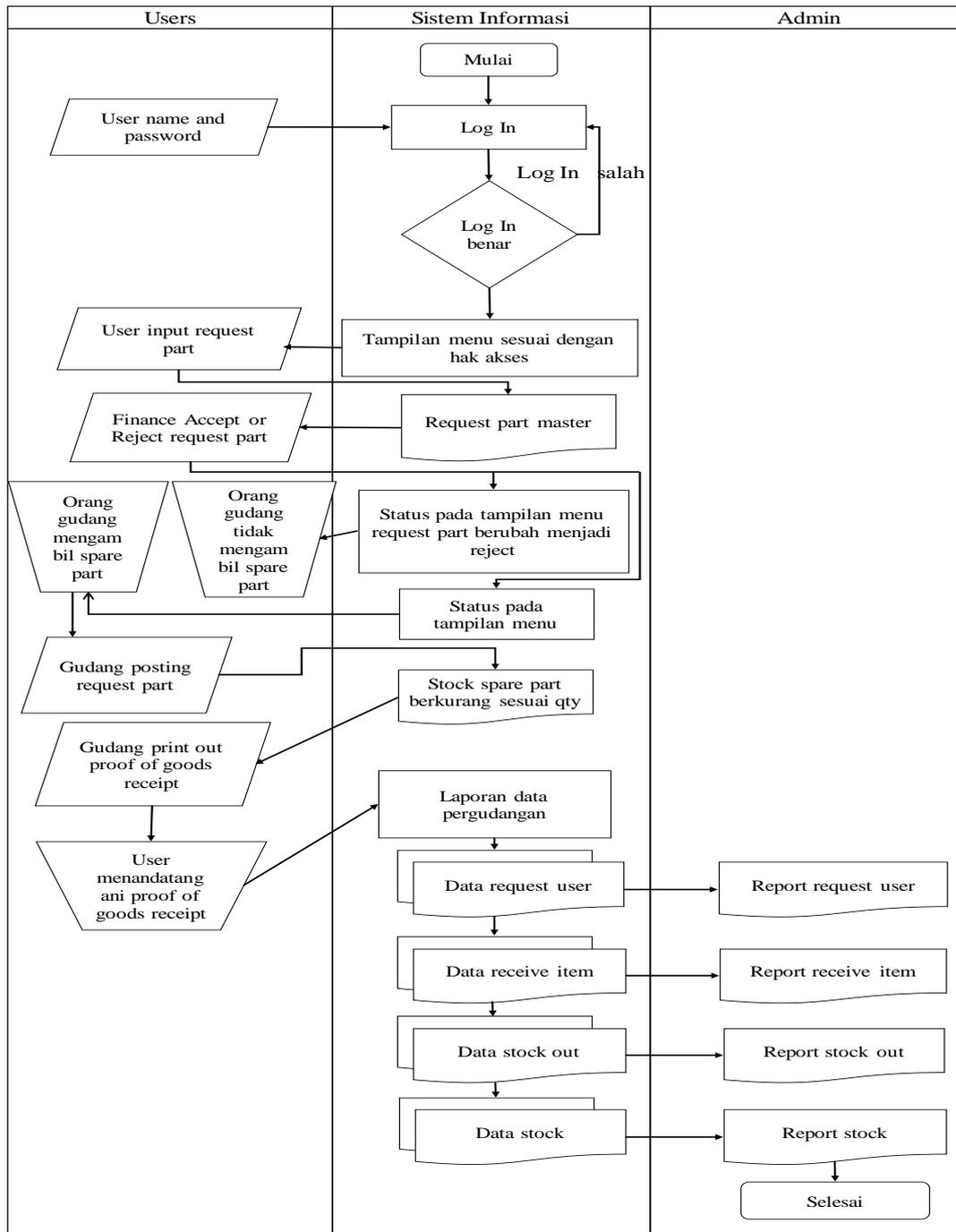
Gambar 4. Old document flow diagram

4.2.3. Logical Design (Data Flow Diagram)

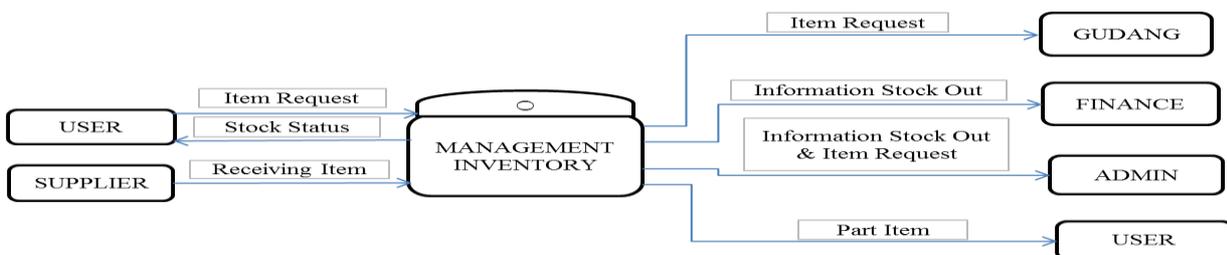
Hasil pengamatan yang sudah dilakukan dari proses pemesanan spare part di CV. ABB saat ini, maka dibuatlah *logical design* menggunakan *data flow diagram*. Gambar 6 menunjukkan bahwa pembuatan DFD yang pertama kali dibuat yaitu diagram konteks. Entity pada diagram konteks terdiri dari entity masukan yaitu user dan supplier, sedangkan entity keluaran yaitu gudang, finance, admin, dan user. Diagram tersebut menjelaskan dari user melakukan request item, melalui proses manajemen pergudangan user mendapatkan info tentang stock status, jika stock yang direquest ada maka user melakukan request item dan info request item masuk ke gudang agar orang gudang mengambilkan item yang direquest. Setelah item diambil oleh orang gudang maka

stock yang ada di gudang berkurang, info stock out dan item request masuk diberikan kepada finance dan admin, untuk item yang sudah direquest diberikan kepada user.

Gambar 7 menjelaskan *user* melakukan *item request* pada proses *request part* yang dibutuhkan untuk memperbaiki mobil, *submitted item request* lalu mengecek *stock status*, kemudian muncul info *availability* dari *stock control* pada tampilan *request part* dan info *stock status* diterima oleh *user*. *Stock control* menerima *submitted item request* dari *user*, data *stock control* dapat dilihat oleh gudang untuk mengecek *stock* yang ada di gudang dengan yang ada di aplikasi. *Finance* dapat melihat data *stock control* untuk mengecek *stock* yang ada di gudang, *stock part* yang sudah minim dan harus dibeli.



Gambar 5. New document flow diagram



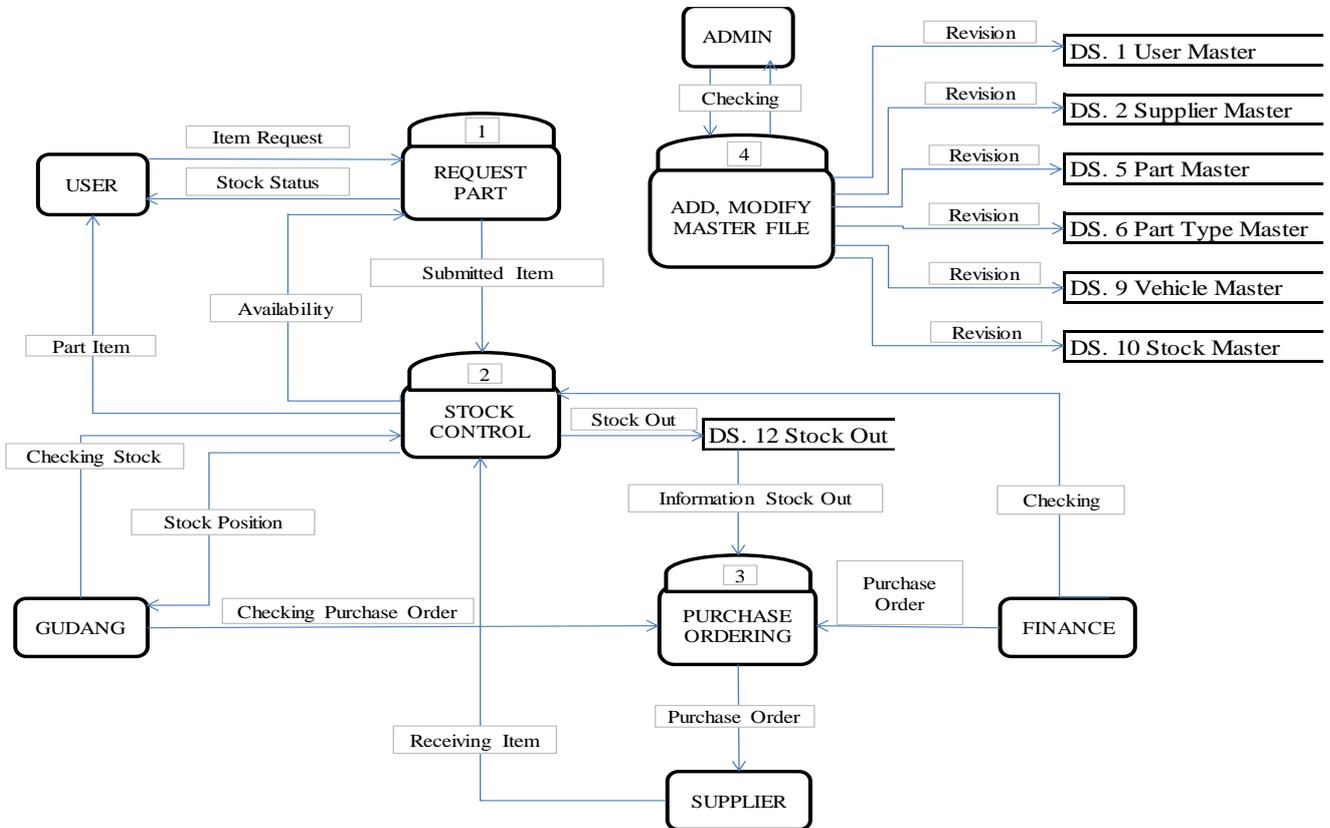
Gambar 6. Diagram konteks DFD

Supplier melakukan pengiriman part ke dalam gudang, sehingga stock yang ada di gudang bertambah. Dalam proses stock control

mengalirkan data stock position kepada gudang untuk memudahkan dalam mengambil part yang direquest oleh user,

stock control memberikan aliran data *availability* di layar tampilan *request part*, proses *stock control* mengalirkan data *stock out* (*spare part* habis) untuk disimpan di DS. 12 *stock out* dan mengalirkan *part item* ke *user*. Dalam proses pembuatan *purchase ordering* mendapatkan aliran data atau informasi *stock out* dari DS. 12 *stock out* untuk mengetahui *stock part* apa saja yang sudah habis dan harus dibeli, *finance* mengalirkan data *purchase order*, setelah *purchase ordering* selesai dibuat maka

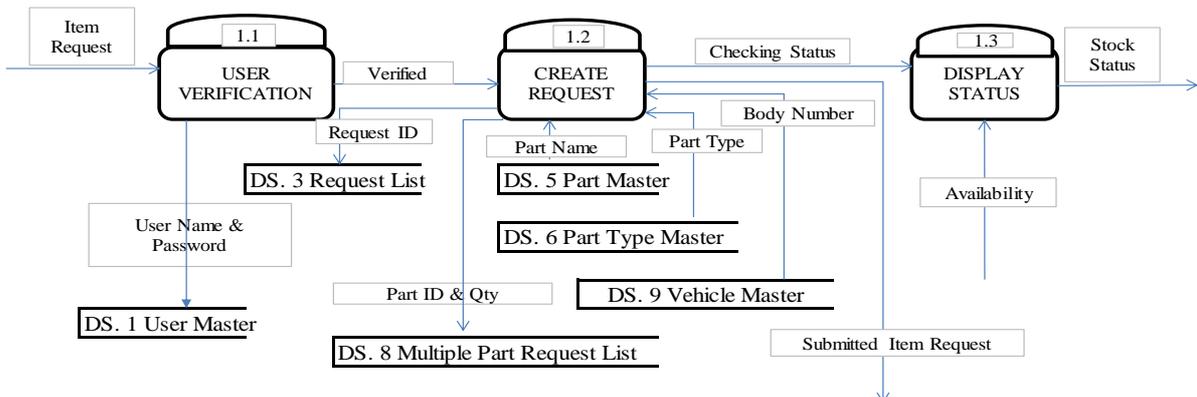
data tersebut dialirkan ke *supplier*. Setelah data masuk ke *supplier* maka *supplier* mengirimkan *spare part* dan orang gudang mengecek data antara *spare part* yang diantar dengan *purchase order* apakah sudah sama atau ada yang kurang atau lebih. *Admin* memiliki wewenang untuk melakukan pengecekan apakah *master file* perlu *add*, dan *modify master file*, jika perlu maka dilakukan *revision* terhadap *master file*.



Gambar 7. Diagram O DFD

Gambar 8 adalah *child diagram* dari diagram 1 proses 1 *request part*. *Child diagram* merupakan rincian proses yang dilakukan dalam proses *request part* dan tidak memiliki *entity* masuk maupun *entity* keluar yang ada hanya proses yang dilakukan, aliran proses atau aliran data, dan data *store*. Dalam melakukan proses *user verification*, mencocokkan *user name* dan *password* yang ada di DS. 1 *user master*, setelah proses *user verification* selesai maka melanjutkan ke proses *create request*. Dalam proses *create request* mengisi *request ID* lalu data mengalir ke DS. 3

request list, jika *user* akan *request part* lebih dari satu maka mengalirkan data ke DS. 8 *multiple part request list* untuk mengetik *request* sesuai kebutuhan. Untuk mencocokkan *part name* maka dibutuhkan aliran dari DS. 5 *part master* dengan *part type* dari DS. 6 *part type master*, untuk mengetahui *user* melakukan pengambilan *item request* untuk mobil apa maka mendapat aliran data dari DS. 9 *vehicle master* yang berisi *body number* atau identitas mobil.



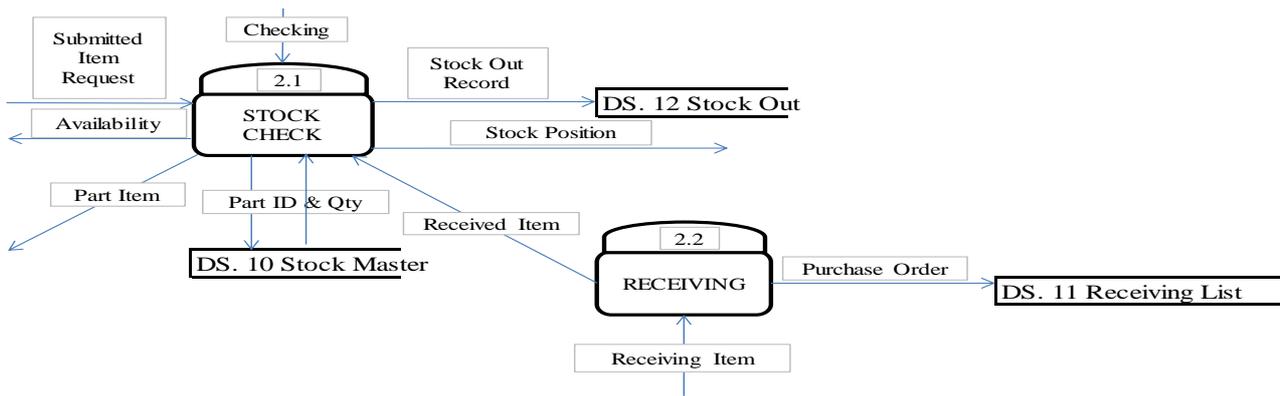
Gambar 8. Diagram 1 Proses 1 Request Part

Pada Gambar 9 dapat dilihat proses *stock check* mendapat aliran data *submitted item request* dari *user*, *checking* dari *finance* untuk mengetahui apakah masih ada *stock* di gudang atau tidak dan mengalirkan data *availability* untuk memberikan info apakah *request item* dari *user* masih ada atau tidak, dan mengalirkan *part item* untuk dilakukan pengecekan oleh *user*. Jika sudah melewati proses *stock check* data *part ID* dan *qty* disimpan di DS. 10 *stock master* dan *stock master* juga bisa mengalirkan data *part ID* dan *qty*, pada proses *stock check* mengalirkan atau *stock out record* data untuk di simpan di DS. 12 *stock out*, dan mengalirkan informasi *stock position* ke gudang agar orang gudang tahu letak *item* yang *direquest*. Dalam proses *stock check* juga mendapatkan aliran data *received item* untuk mengecek *part* yang dikirim oleh *supplier* sudah sama atau belum dengan *purchase order*. Proses *receiving* dimulai dari *receiving item* yang dikirim oleh *supplier* yang diterima oleh gudang lalu orang gudang mengecek kondisi semua item dan menyimpan *purchase order* pada DS. 11 *receiving list*.

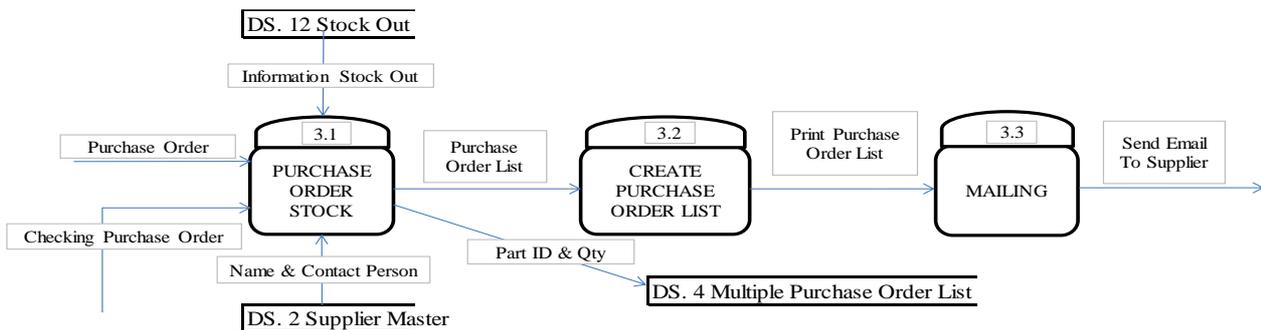
Pada Gambar 10 diagram 1 proses 3 yaitu proses *purchase order* terdapat proses *purchase order stock* yang datanya bisa dicocokkan dengan *purchase order* yang telah dibuat oleh *finance*,

mendapatkan aliran informasi *stock out* dari DS. 12 *stock out* untuk mengecek *stock* yang sudah tidak ada di gudang, mendapatkan aliran data *name* dan *contact person supplier* dari DS. 2 *Supplier master* untuk memilih *supplier* yang sesuai dengan *purchase order* yang sedang dibuat, lalu data di simpan pada DS. 4 *multiple purchase order list* seperti data *part ID* dan *qty* yang akan diorder. Setelah proses *purchase order stock* selesai maka selanjutnya mengalirkan data *purchase order list* ke proses *create purchase order list*, lalu *print purchase order list* dan ditanda tangani oleh *finance* atau *admin* bukti bahwa *finance* atau *admin* menyetujui *purchase order* yang telah dibuat. Proses terakhir *e-mail purchase order list* ke *supplier* agar proses selanjutnya bisa berjalan dengan lancar.

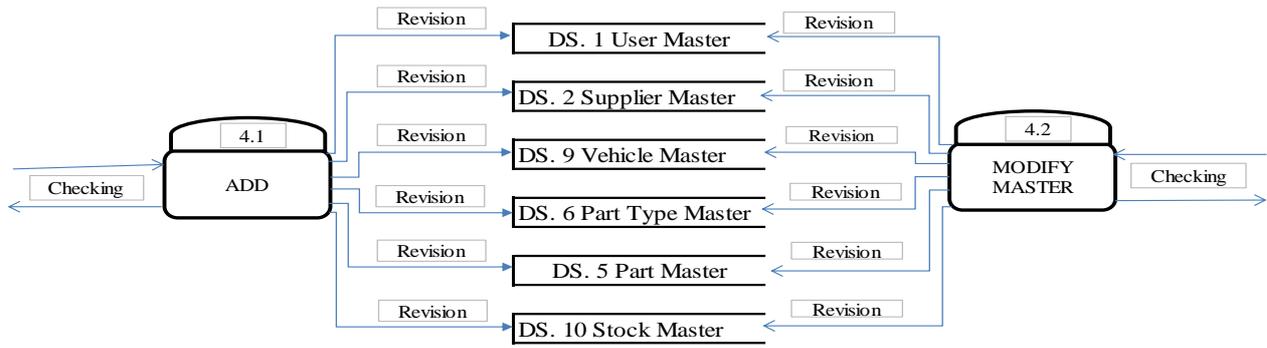
Pada Gambar 11 terlihat *child diagram* proses *add, modify master file, checking* hanya bisa dilakukan oleh *admin* karena kedudukan *admin* lebih tinggi dari *entity* yang lain. Semua data bisa dilakukan *add* dan untuk *modify master file* dapat dilakukan pada semua *master file*. Semua *master file* tidak bisa dihapus karena jika salah satu data dihapus dapat menyebabkan hilangnya data pada *report*.



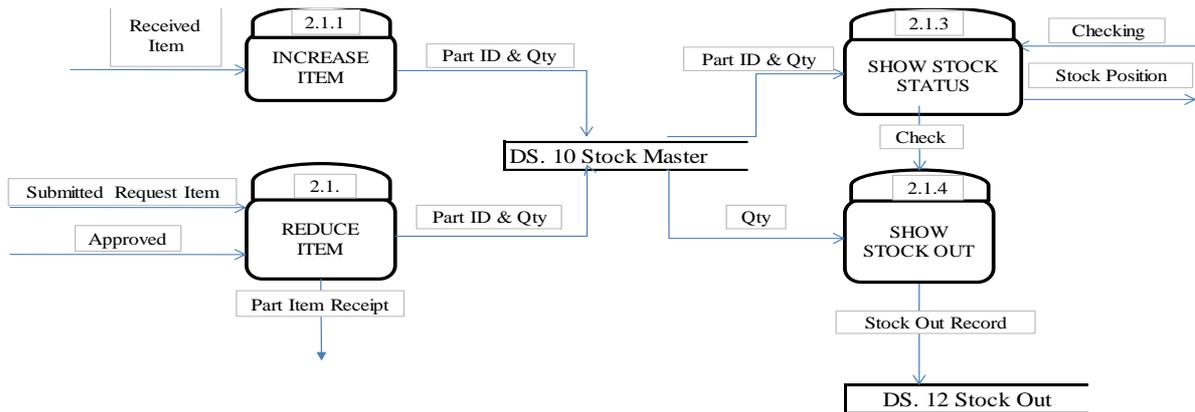
Gambar 9. Diagram 1 Proses 2 Stock Control



Gambar 10. Diagram 1 Proses 3 Purchase Order



Gambar 11. Diagram 1 Proses 4 Add, Modify Master File



Gambar 12. Diagram 2 Proses 2.1 Stock Check

4.2.4. Physical Design (Entity Relationship Diagram)

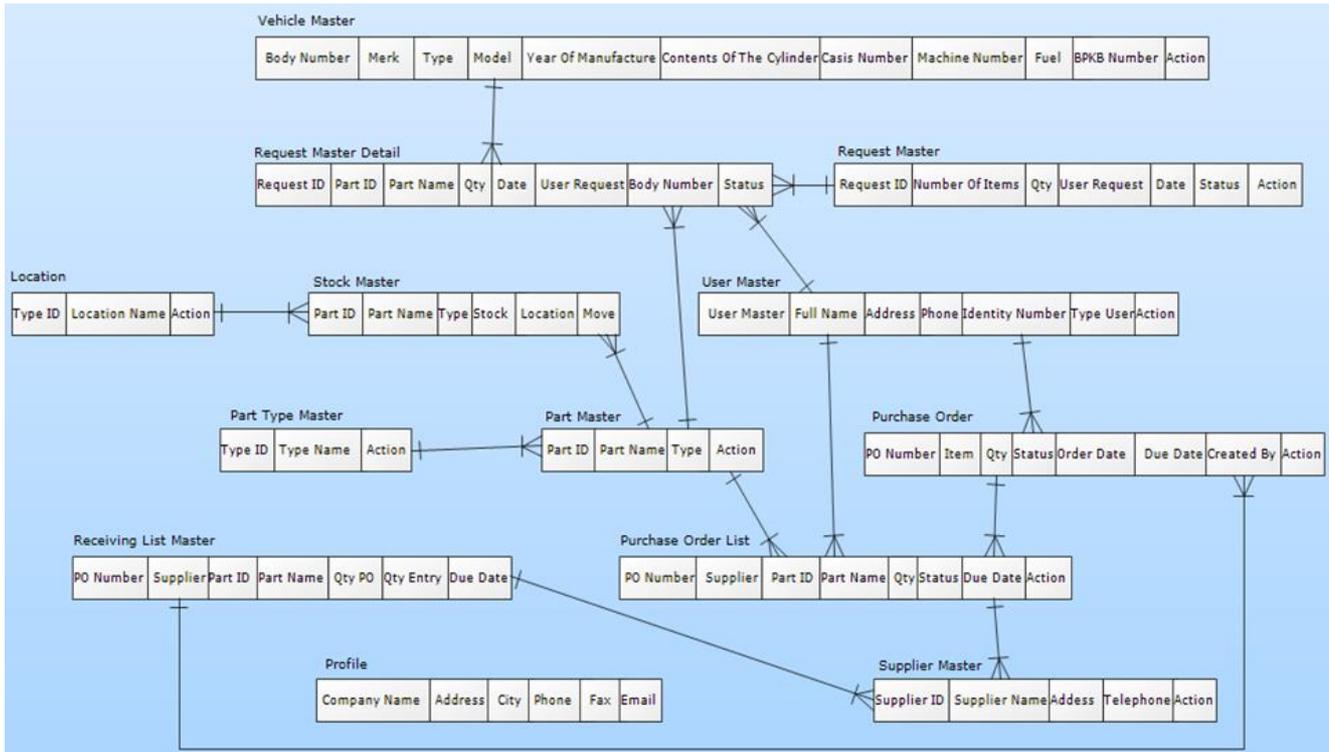
Physical design pada vehicle master terdapat tabel atau entitas body master number, merk, type, model, year of manufacture, contentents of the cylinder, casis number, machine number, fuel, BPKB, dan action yang berelasi dengan request master detail satu ke banyak. Request master detail berelasi dengan request master, user master dan part master satu ke banyak, request master detail terdapat tabel request ID, part ID, part name, qty, date, user request, body number, dan status. Pada request master terdapat tabel request ID, number of items, qty, user request, date, status, dan action. Location berelasi satu ke banyak dengan stock master, tabel type ID, location name, dan action merupakan tabel dari location. Stock master terdapat tabel part ID, part name, type, stock, location, dan move, stock master berelasi dengan part master yang memiliki tabel part ID, part name, type, dan action. User master berelasi dengan purchase order dan purchase order list satu ke banyak, pada user master terdapat tabel user master, full name, address, phone, identity number, type user dan action. Part type master berelasi dengan part master satu ke banyak, dengan tabel type ID, type name, dan action. Part master berelasi satu ke banyak dengan purchase order list, pada part master terdapat tabel part ID, part name, type, dan action. Purchase order berelasi satu ke banyak terhadap purchase order dan berelasi banyak ke satu dengan receiving list master, pada purchase order terdiri dari tabel PO number, item, qty, status, order date, due date, created by, dan action. Pada purchase order list terdapat tabel PO number, supplier, part ID, part name, qty, status, due date, action, dan berelasi dengan supplier master. Untuk receiving master berelasi satu ke banyak dengan supplier

master dan purchase order, dalam receiving list master terdapat tabel PO number, supplier, part ID, part name, qty PO, qty entry, dan due date. Pada supplier master terdapat supplier ID, supplier name, address, telephone, dan action. Pada physical design hanya ada satu menu yang tidak berelasi dengan menu yang lain yaitu profile yang terdapat tabel company name, address, city, phone, fax, dan email. Gambar 13 menunjukkan gambar physical design.

4.2.5. Identifikasi Kelemahan Sebelum Adanya Sistem Informasi

Pada sistem pergudangan sebelum adanya sistem informasi memiliki beberapa kekurangan yaitu:

- Dari segi user tidak tahu spare part yang ada di gudang ada atau tidak, merasa dirugikan karena untuk melakukan pengambilan spare part dari gudang, tidak adanya bukti pengambilan spare part dari user.
- Dari segi gudang, orang gudang lebih mudah melakukan manipulasi data terhadap data pemesanan spare part, spare part masuk, spare part keluar, stock spare part dan bon palsu. Orang gudang tidak mau membuat laporan spare part masuk dan stock spare part dengan alasan banyaknya pekerjaan dan pemesanan spare part dari user, sehingga orang gudang beralasan tidak sempat untuk membuat laporan spare part masuk dan stock spare part. Orang gudang membuat pencatatan spare part keluar secara manual seperti ditulis tangan di buku dan lembaran kertas, sehingga sangat rentan terjadinya kesalahan pencatatan.



Gambar 13. Physical diagram (ERD)

- c. Dari segi *finance* masih membuat *purchase order* secara manual dan kesulitan dalam melakukan pengawasan spare part mulai dari *report stock out*, *report stock*, *report receive item* dan *report user*.
- d. Dari segi *admin* mendapat kesulitan dalam melakukan pengawasan *spare part* yang ada di gudang CV. ABB.

4.3. Design

Dalam pembuatan sistem informasi tools yang digunakan yaitu *document flow diagram*, *data flow diagram*, dan *software* yang digunakan untuk pembuatan analisis kebutuhan yaitu menggunakan *case tools* dan *Microsoft visio*. Sedangkan untuk *design* informasi melalui *input/output design*, *database design software* yang digunakan yaitu *PHP7 & MySQL*, dan untuk pembuatan ERD menggunakan *software case tools*.

4.3.1. Input/Output Design

Hasil pembuatan *design input* dan *output* yaitu sebagai berikut:

a. Form Request Part Design

Pada *design request part* berisi request ID yang otomatis ada dan tidak bisa dirubah, part ID bisa diketik huruf depannya manual lalu muncul kumpulan part ID dan pilih yang sesuai dengan keinginan, *part detail* otomatis muncul jika part ID sudah diisi, *body number* diisi untuk mengetahui part untuk memperbaiki mobil dengan *body number* berapa, Qty bisa diisi sesuai kebutuhan part yang akan digunakan, setelah selesai pengisian lalu klik tombol *Add*. Pada tampilan dibawahnya dapat dilihat *request part* yang sudah diisi, jika *request* lebih dari satu jenis part maka isi kembali *request part* pada tampilan diatas, tetapi jika sudah selesai akan muncul tombol *posting* dibawah tombol *Add*.

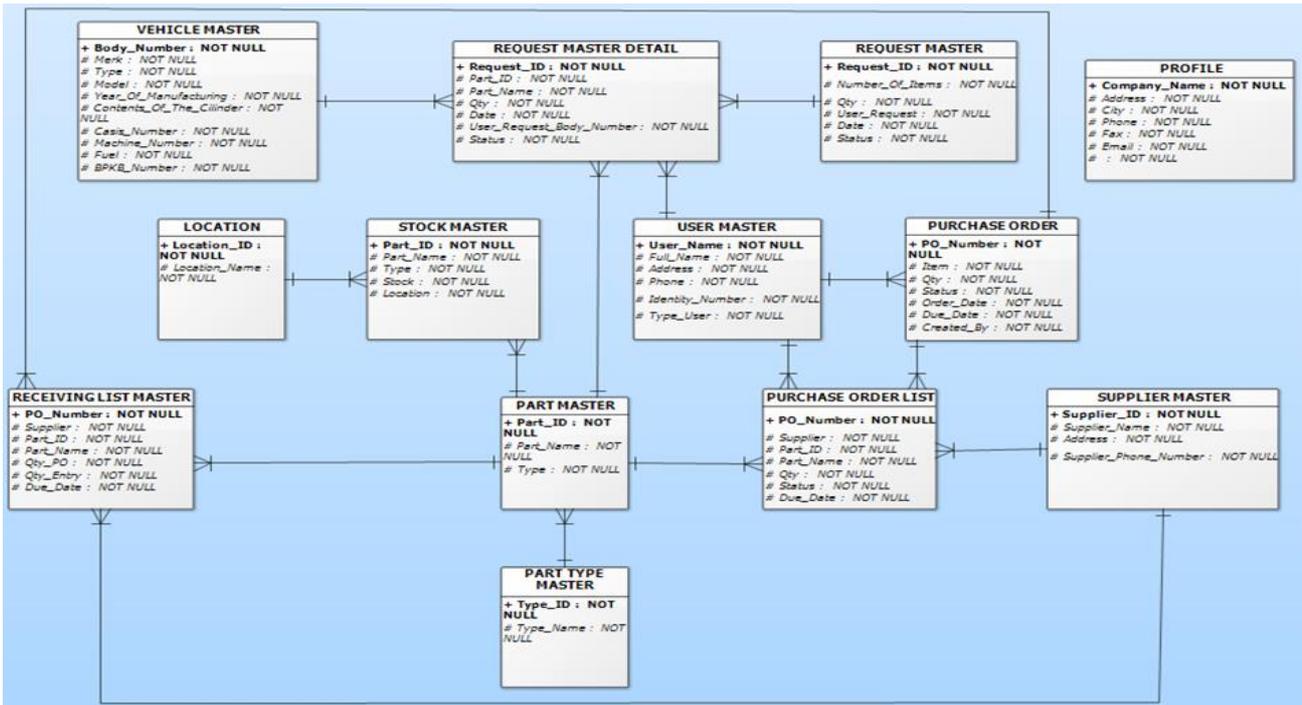
Gambar 14. Form request part design

b. Proof of Goods Reveipt Design

Design ini digunakan untuk melakukan serah terima part antara orang gudang dengan user (orang bengkel).

c. Create Purchase Order Design

Ketika orang *finance* akan membuat *purchase order* maka bisa klik pada menu *purchase order*, klik PO, lalu klik *create purchase order*, dan muncul tampilan pada gambar dibawah. Data yang diperlukan dalam pembuatan *purchase order* yaitu *part name*, nama *supplier*, *qty*, *due date* sedangkan data yang lainnya otomatis muncul setelah data yang lain di isi. Jika sudah selesai mengisi *purchase order* maka klik *save*, dan muncul *order* yang ditulis pada layar yang dibawah. Jika ingin *order part* lebih dari satu jenis maka tulis lagi *part name*, nama *supplier*, *qty* dan *due date* yang dibutuhkan lalu *posting*. Yang bisa melakukan *create purchase order* yaitu *finance* dan *admin*.



Gambar 18. Database design

4.3.3. Tampilan Database Design

Database merupakan sekumpulan informasi yang disimpan di dalam komputer secara sistematis sehingga dapat diakses menggunakan suatu program komputer untuk memperoleh informasi dari database tersebut. Berikut kumpulan database design yang ada di aplikasi:

a. Home Design

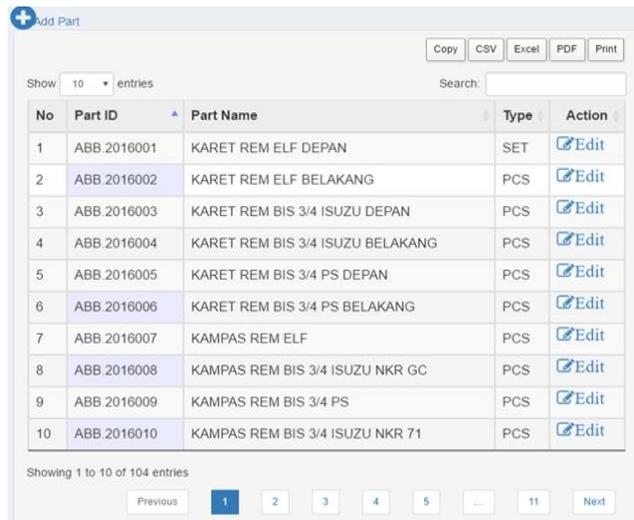
Pada tampilan home terlihat peringatan tentang stock yang kurang dari tiga dan harus segera dilakukan order agar request dari user bisa tercukupi. Tampilan ini bisa dilihat oleh semua pengguna yaitu user, gudang, finance, dan admin agar bisa saling mengingatkan dan segera dilakukan purchase order ke supplier.



Gambar 19. Home Design

b. Part Master Design

Part master digunakan untuk mengetahui part apa saja yang diperlukan untuk memperbaiki mobil.



Gambar 20. Part master design

Pada Gambar 20 dalam aplikasi yang bisa melihat part master yaitu user untuk melakukan request part, gudang untuk melihat jenis - jenis part yang ada di gudang, finance untuk melakukan purchase order, dan admin untuk melakukan pengontrolan part yang ada di gudang. Pada part master berisi informasi tentang part ID, part name dan type.

c. Stock Master Design

Stock master digunakan untuk mengetahui stock part yang masih ada di gudang. Yang bisa melihat stock master yaitu user untuk melihat stock request part yang masih ada di gudang, gudang melakukan pengecekan part, finance mengecek part apa saja yang masih ada di gudang, dan admin untuk mengontrol stock part yang masih ada di gudang. Pada stock master berisi informasi seperti part ID, part name, type, stock, dan location.

Part ID	Part Name	Type	Stock	Location	Move
ABB.2016001	KARET REM ELF DEPAN	SET	49	A01	Move
ABB.2016002	KARET REM ELF BELAKANG	PCS	50	A02	Move
ABB.2016003	KARET REM BIS 3/4 ISUZU DEPAN	PCS	49	A03	Move
ABB.2016004	KARET REM BIS 3/4 ISUZU BELAKANG	PCS	50	A04	Move
ABB.2016005	KARET REM BIS 3/4 PS DEPAN	PCS	50	A05	Move
ABB.2016006	KARET REM BIS 3/4 PS BELAKANG	PCS	50	A06	Move
ABB.2016007	KAMPAS REM ELF	PCS	10	A07	Move
ABB.2016008	KAMPAS REM BIS 3/4 ISUZU NKR GC	PCS	10	A08	Move
ABB.2016009	KAMPAS REM BIS 3/4 PS	PCS	10	A09	Move
ABB.2016010	KAMPAS REM BIS 3/4 ISUZU NKR 71	PCS	10	A10	Move

Gambar 21. Stock master design

d. User Master Design

User master hanya bisa dilihat oleh finance dan admin saja untuk mengetahui siapa saja yang menggunakan aplikasi dan sebagai apa orang tersebut, namun finance hanya bisa add user saja sementara admin bisa add user sekaligus edit. Pada user master berisi tentang user name, full name, address, phone, identity number, type user, dan action.

User Name	Full Name	Address	Phone	Identity Number	Type user	Action
gudang	Bagian Gudang 2	JL xxx Nomor 23	098887772999	123211123	2	
gudang-hery	hery azwir	jalan	007	1020	2	
finance	finance	finance	111111	125444220	3	
finance-hery	hery azwir	Jalan	007	1020	3	
finance1	finance1	hijkl	6788	4454678	3	
user	user	user	02109988	0198828877	4	
user-hery	hery azwir	Jalan	007	1020	4	
user1	user1	address	56789	9878665	4	

Gambar 22. User master design

e. PO Master Design

Berikut merupakan gambar PO master design:

PO Number	Item	Qty	Status	Order Date	Due Date	Created By	Action
PO20160005	1	1	NEW	2016-11-29	2016-12-10	admin	
PO20160006	1	12	NEW	2016-11-29	2016-11-30	finance	
PO20160001	1	50	CLOSE	2016-11-26	0000-00-00	admin	
PO20160002	5	250	CLOSE	2016-11-26	0000-00-00	admin	
PO20160003	4	40	CLOSE	2016-11-26	0000-00-00	admin	
PO20160004	1	20	CLOSE	2016-11-27	0000-00-00	finance	

Gambar 23. PO master design

PO master adalah kumpulan purchase order yang sudah dibuat oleh finance atau admin. pada PO master terdapat informasi tentang PO number, item, status, order date, due date create by (dibuat oleh, bisa admin atau finance), action. Untuk melakukan purchase order bisa tekan tombol create purchase order pada bagian atas sebelah kiri.

f. Receive Design

Tampilan ini muncul ketika orang gudang, finance, ataupun admin melakukan good receive, pada tampilan ini terlihat qty entry yang diterima berapa dan due date. Sehingga dapat diketahui penambahan stock dari datangnya part yang diorder itu berapa. Jika sudah selesai maka posting agar stock langsung bertambah di gudang

No	PO Number	Supplier	Part ID	Part Name	QTY PO	QTY Entry	Due Date
1	PO20160005	VEN.6006	ABB.2016006	KARET REM BIS 3/4 PS BELAKANG	1		10 Desember 2016

Gambar 24. Receive Design

4.4. Analisis Perbandingan

Dari implementasi yang telah dilakukan dilakukan analisis perbandingan pada Tabel 3.

Tabel 3. Analisis perbandingan

No.	Tanpa Sistem Informasi Pergudangan	Dengan Sistem Informasi Pergudangan
1	Orang gudang budang biasanya membeli barang atau spare part yang stoknya sudah habis tanpa adanya perintah dari atasan dan tidak menginformasikan stok spare part apa saja yang sudah habis	Dalam pembelian atau order, dan pembayaran dilakukan oleh finance untuk menghindari bon palsu
2	Orang gudang satang ke finance memberikan bon pembelian spare part dengan harga yang jauh berbeda dengan alasan pembelian tempat pembelian spare part yang biasa masih tutup jadi membeli spare part di supplier yang lain	Orang gudang sudah tidak boleh memberikan bon pembelian spare part, orang gudang hanya bertugas memberikan spare part untuk user dan menempatkan spare part yang dikirim dari supplier
3	User (orang bengkel) datang ke gudang untuk melakukan pemesanan spare part	User bisa datang langsung ke gudang atau melakukan request part melalui sistem informasi pergudangan
4	Adanya peluang manipulasi data	Memperkecil terjadinya manipulasi data

Tabel 3. Analisis perbandingan (Lanjutan)

No.	Tanpa Sistem Informasi Pergudangan	Dengan Sistem Informasi Pergudangan
4	Adanya peluang manipulasi data	Memperkecil terjadinya manipulasi data
5	Orang gudang mencatat manual <i>spare part</i> yang diambil oleh <i>user</i>	Orang gudang sudah tidak menulis manual di kertas cukup klik posting pada sistem informasi pergudangan
6	Tidak ada laporan <i>spare part</i> masuk	Adanya laporan <i>spare part</i> masuk
7	Sulit dilakukan pengontrolan dan pengecekan <i>spare part</i> masuk dan keluar	Mudah dilakukan pengontrolan dan pengecekan <i>spare part</i> masuk dan keluar

5. KESIMPULAN DAN SARAN

Dari penelitian (studi kasus) ini didapatkan beberapa kesimpulan yaitu:

1. Sistem informasi pergudangan yang diimplementasikan pada CV. ABB berhasil secara cukup efektif dalam menekan permasalahan yang terjadi di manajemen gudang selama ini.
2. Dengan adanya sistem informasi manajemen pergudangan di CV. ABB, manajemen gudang menjadi lebih akuntabel dan transparan.

Dari penelitian ini didapatkan saran untuk pengembangan sistem informasi selanjutnya yaitu:

1. Adanya fitur notifikasi jika ada STNK yang akan habis masa berlakunya, sehingga dapat terhindar dari masalah telat perpanjang STNK yang dapat menambah pengeluaran karena harus membayar denda pajak.
2. Untuk *finance* bisa ditambahkan menu harga dari setiap *spare part* untuk memudahkan dalam pemesanan *spare part*, menu *report finance* yang berisi rincian pengeluaran pembelian *spare part*.

DAFTAR PUSTAKA

- [1] S. Emmett. *Excellence in Warehouse Management: How to Minimise Cost and Maximises Value*. West Sussex: John Wiley and Sons, 2005.

- [2] M.t. Hompel dan T. Schmidt. *Warehouse Management*. Dortmund: Springer-Verlag, 2007.
- [3] H.I. Yunarto dan M.G. Santika. *Business Concept Implementation Series In Inventory Management*. Jakarta: Elex Media, 2005.
- [4] J. Hartono. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Penerbit Andi, 2005.
- [5] Al Fatta, Hanif. *Analisis dan Perancangan Sistem Informasi*. Yogyakarta: Penerbit Andi, 2007.
- [6] T. Sutarbi. *Analisis Sistem Informasi*. Yogyakarta: Penerbit Andi, 2004.
- [7] H.I. Pohan. *Pengantar Perancangan Sistem Informasi*. Jakarta: Penerbit Erlangga, 1997.
- [8] D.M. Kroenke. *Database Processing: Fundamentals, Design and Implementation, Seventh Edition*. United State Of America: Prentice Hall, 2000.
- [9] T.M. Connolly dan C.E. Begg. *Database system: a Practical Approach To Design, Implementation, and Management*. America: Pearson Education, 2010.
- [10] Anhar. *PHP & MySQL Secara Otodidak*. Jakarta: PT. Trans Media, 2010.

BIODATA PENULIS



Hery Hamdi Azwir

Hery Hamdi Azwir adalah dosen di Universitas Presiden, Program Studi Teknik Industri. Lahir di Bandung dan meraih gelar Sarjana Teknik Elektro dari Institut Teknologi Bandung (ITB) dengan konsentrasi pada Sistem Komputer, dan Magister Teknik dalam Departemen Teknik dan Manajemen Industri juga di ITB.



Oktavia Patriani

Oktavia Patriani adalah mahasiswi di Universitas Presiden, Program Studi Teknik Industri. Penulis lahir di Banyumas Jawa Tengah.