

BAB II

KAJIAN LITERATUR

2.1 Sistem Informasi Inventaris

Sistem informasi inventaris merupakan sistem yang dirancang untuk mendukung proses pengelolaan persediaan barang dalam suatu organisasi. Sistem ini berfungsi untuk mencatat data barang masuk dan keluar, memantau ketersediaan stok, serta menyajikan informasi inventaris secara akurat dan terstruktur guna mendukung kegiatan operasional dan pengambilan keputusan manajemen. Dalam konteks usaha material bangunan, sistem informasi inventaris memiliki peran penting karena tingginya variasi jenis barang, volume stok, serta kebutuhan pencatatan yang akurat dan *real-time* [8].

Dalam praktiknya, sistem informasi inventaris tidak hanya berfungsi sebagai alat pencatatan data, tetapi juga sebagai sarana pengendalian persediaan yang memungkinkan perusahaan melakukan *monitoring* terhadap kondisi stok secara lebih sistematis. Dengan adanya sistem ini, organisasi dapat mengetahui jumlah barang yang tersedia, lokasi penyimpanan barang, serta riwayat transaksi yang berkaitan dengan setiap *item* persediaan.

Dalam konteks usaha material bangunan, Penerapan sistem informasi inventaris bertujuan untuk meminimalkan kesalahan pencatatan, meningkatkan efisiensi kerja, serta memastikan ketersediaan barang sesuai dengan kebutuhan *customer*. Dengan sistem yang terkomputerisasi, proses pengelolaan persediaan menjadi lebih terkontrol dan transparan dibandingkan dengan pencatatan manual [9][10].

2.1.1 Manajemen Inventaris

Manajemen inventaris merupakan proses perencanaan, pengorganisasian, pengendalian, dan pengawasan terhadap persediaan barang agar jumlah stok yang tersedia tetap sesuai dengan kebutuhan operasional perusahaan. Persediaan barang menjadi salah satu aset penting dalam kegiatan bisnis karena berhubungan langsung dengan proses penjualan, pelayanan *customer*, dan kelancaran aktivitas operasional perusahaan. Oleh karena itu, perusahaan perlu menerapkan manajemen inventaris yang baik agar dapat menjaga keseimbangan antara ketersediaan stok dan efisiensi biaya penyimpanan barang.

Dalam kegiatan operasional perusahaan, inventaris mencakup seluruh barang yang disimpan untuk mendukung proses bisnis, baik berupa bahan baku, barang setengah jadi, maupun barang jadi yang siap dipasarkan kepada *customer*. Menurut penelitian [11], pengelolaan inventaris yang baik mampu meningkatkan efisiensi operasional perusahaan

karena seluruh data barang dapat dipantau dan dikendalikan secara lebih terstruktur. Selain itu, manajemen inventaris juga membantu perusahaan dalam mengurangi risiko kekurangan stok (*stockout*) maupun penumpukan barang (*overstock*) yang dapat menyebabkan kerugian operasional.

Tujuan utama manajemen inventaris adalah memastikan bahwa jumlah stok barang selalu tersedia sesuai kebutuhan tanpa menyebabkan pemborosan biaya penyimpanan. Untuk mencapai tujuan tersebut, perusahaan perlu melakukan pengendalian terhadap arus barang masuk dan barang keluar secara sistematis. Pengendalian inventaris meliputi proses pencatatan data barang, pemantauan jumlah stok, pengelompokan kategori produk, hingga penyusunan laporan inventaris sebagai dasar pengambilan keputusan manajemen.

Banyak usaha kecil dan menengah (UMKM), pengelolaan inventaris masih dilakukan secara manual menggunakan pencatatan tertulis atau *spreadsheet* seperti Microsoft Excel. Penggunaan Excel memang dinilai lebih mudah dan murah untuk diterapkan, namun metode tersebut memiliki berbagai keterbatasan. Proses pembaruan stok harus dilakukan secara manual setiap kali terjadi transaksi barang masuk maupun barang keluar. Selain itu, data inventaris biasanya disimpan pada beberapa lembar kerja yang terpisah sehingga proses pencarian dan pencocokan data membutuhkan waktu yang lebih lama. Kondisi tersebut meningkatkan risiko terjadinya human error, keterlambatan pembaruan data, dan ketidaksesuaian antara data stok dalam sistem dengan kondisi fisik barang di gudang.

Permasalahan penggunaan Excel dalam pengelolaan inventaris juga dijelaskan dalam penelitian [12] yang menyatakan bahwa sistem inventaris berbasis *spreadsheet* menyebabkan proses rekapitulasi data stok menjadi lambat dan kurang efisien karena seluruh proses pencatatan dilakukan secara manual. Selain itu, penelitian tersebut menunjukkan bahwa keterlambatan pembaruan data inventaris dapat memengaruhi kualitas pengambilan keputusan dan pelayanan kepada *customer*.

Perkembangan teknologi informasi mendorong perusahaan untuk beralih dari sistem manual menuju sistem informasi inventaris berbasis komputer dan *web*. Sistem informasi inventaris berbasis *web* memungkinkan proses pencatatan, pengelolaan, dan *monitoring* stok dilakukan secara otomatis dan *real-time*. Setiap transaksi barang masuk dan barang keluar dapat langsung memperbarui jumlah stok dalam *database* sehingga data inventaris selalu akurat dan mudah diakses oleh pengguna sistem. Selain itu, sistem berbasis *web* juga memungkinkan integrasi antarbagian sehingga seluruh data inventaris tersimpan dalam satu sistem yang terpusat.

Dalam penelitian ini, manajemen inventaris diterapkan melalui pengembangan sistem informasi material bangunan berbasis *web* pada Toko Elegant Keramik. Sistem yang dikembangkan dirancang untuk membantu *admin* dalam mengelola kategori produk, data barang, stok, serta transaksi barang masuk dan barang keluar secara otomatis dan *real-time*. Selain itu, sistem juga menyediakan tampilan *customer* yang memungkinkan *customer* melihat informasi produk dan ketersediaan stok secara mandiri tanpa perlu menghubungi *admin* melalui *WhatsApp*. Dengan demikian, penerapan sistem informasi inventaris berbasis *web* diharapkan mampu meningkatkan efisiensi pengelolaan stok, menjaga akurasi data inventaris, dan meningkatkan kualitas pelayanan kepada *customer*.

2.1.2 Akurasi Data Inventaris

Akurasi data inventaris merupakan tingkat kesesuaian antara jumlah stok yang tercatat dalam sistem dengan kondisi fisik barang di gudang. Tingkat akurasi inventaris yang rendah dapat menyebabkan kesalahan dalam pengambilan keputusan, seperti kesalahan pemesanan barang, keterlambatan pelayanan *customer*, serta meningkatnya biaya operasional. Oleh karena itu, sistem informasi inventaris harus mampu mendukung pencatatan stok secara konsisten dan *real-time* guna menjaga akurasi data persediaan [13].

Data inventaris digunakan perusahaan untuk berbagai kebutuhan operasional, seperti proses pembelian barang, pengendalian stok, penyusunan laporan inventaris, hingga pelayanan kepada *customer*. Oleh karena itu, ketidaksesuaian data stok dapat menimbulkan berbagai dampak negatif, seperti kesalahan pemesanan barang, keterlambatan pelayanan *customer*, penumpukan stok, hingga meningkatnya biaya operasional perusahaan. Selain itu, informasi stok yang tidak akurat juga dapat menyebabkan *customer* memperoleh informasi yang salah mengenai ketersediaan barang sehingga menurunkan tingkat kepuasan *customer*.

Rendahnya akurasi data inventaris umumnya disebabkan oleh proses pencatatan manual yang masih menggunakan *spreadsheet* atau dokumen tertulis. Pada sistem berbasis Excel, proses pembaruan stok dilakukan secara manual oleh *admin* setiap kali terjadi transaksi barang masuk maupun barang keluar. Jika *admin* terlambat memperbarui data atau melakukan kesalahan input, maka jumlah stok yang tercatat dalam sistem tidak akan sesuai dengan kondisi fisik barang di gudang.

Penelitian [14] menjelaskan bahwa penggunaan sistem inventaris manual menyebabkan tingginya risiko ketidaksesuaian data stok akibat keterlambatan pembaruan data dan kesalahan pencatatan. Penelitian tersebut juga menunjukkan bahwa data inventaris yang tidak akurat dapat menghambat proses pengambilan keputusan perusahaan dan menurunkan efisiensi operasional.

Untuk meningkatkan akurasi data inventaris, perusahaan perlu menerapkan sistem informasi inventaris yang mampu melakukan pencatatan dan pembaruan stok secara otomatis dan *real-time*. Sistem inventori berbasis *web* memungkinkan setiap transaksi barang masuk maupun barang keluar langsung memengaruhi jumlah stok yang tersimpan di dalam *database*. Dengan demikian, data inventori yang ditampilkan oleh sistem akan selalu diperbarui secara otomatis dan lebih sesuai dengan kondisi aktual barang di gudang.

Selain itu, sistem inventori modern juga mendukung proses *monitoring* stok secara terintegrasi sehingga *admin* dapat dengan mudah memantau perubahan jumlah barang dan riwayat transaksi inventori. Penggunaan *database* terpusat membantu mengurangi risiko duplikasi data serta meningkatkan konsistensi informasi inventori antarbagian dalam perusahaan. Sistem berbasis *web* juga mempermudah proses pencarian data dan penyusunan laporan stok karena seluruh data tersimpan secara digital dan dapat diakses kapan saja.

Pada penelitian ini, peningkatan akurasi data inventori dilakukan melalui pengembangan sistem informasi material bangunan berbasis web menggunakan ReactJS dan Cloud Firestore. Sistem dirancang agar setiap perubahan stok dapat diperbarui secara otomatis dan *real-time* sehingga informasi persediaan barang menjadi lebih akurat dan mudah diakses oleh *admin* maupun *customer*. Dengan penerapan sistem ini, diharapkan proses pengelolaan inventori di Toko Elegant Keramik menjadi lebih efektif dan mampu meminimalkan risiko kesalahan pencatatan stok.

2.1.3 Lead Time Dalam Manajemen Inventori

Lead Time adalah waktu yang dibutuhkan sejak pemesanan barang dilakukan hingga barang tersebut diterima dan siap digunakan atau dijual. *Lead Time* menjadi faktor penting dalam pengendalian inventori karena berpengaruh langsung terhadap penentuan jumlah persediaan dan waktu pemesanan ulang [15].

Secara umum, *Lead Time* terdiri dari beberapa tahapan, yaitu waktu pemrosesan pesanan, waktu pengiriman barang oleh *supplier*, serta waktu penerimaan dan pengecekan barang di gudang. Setiap *supplier* dapat memiliki *Lead Time* yang berbeda-beda tergantung pada lokasi pengiriman, kapasitas produksi, metode distribusi, dan kondisi transportasi. Oleh karena itu, perusahaan perlu memahami pola *Lead Time* agar dapat menentukan waktu pemesanan ulang barang secara tepat dan efisien.

Dalam pengelolaan inventori, *Lead Time* digunakan sebagai dasar dalam menentukan *Reorder Point* (ROP) atau titik pemesanan ulang barang. Jika perusahaan melakukan pemesanan terlalu lambat, maka stok barang dapat habis sebelum pesanan baru tiba di gudang. Sebaliknya, jika perusahaan melakukan pemesanan terlalu cepat dalam

jumlah besar, maka biaya penyimpanan barang akan meningkat karena terjadi penumpukan stok. Dengan demikian, pengelolaan *Lead Time* yang baik sangat diperlukan untuk menjaga keseimbangan antara ketersediaan stok dan efisiensi biaya operasional.

Pada banyak UMKM dan toko bangunan, proses pemantauan stok dan pemesanan barang masih dilakukan secara manual menggunakan Microsoft Excel. Kondisi tersebut menyebabkan *admin* harus memeriksa data stok secara satu per satu sebelum menentukan kapan harus melakukan pemesanan ulang kepada *supplier*. Selain memerlukan waktu yang lebih lama, metode manual juga meningkatkan risiko keterlambatan pengambilan keputusan akibat data stok yang tidak diperbarui secara *real-time*. Akibatnya, perusahaan lebih rentan mengalami *stockout* atau kekosongan barang yang dapat menghambat pelayanan kepada *customer*.

Penelitian [16] menjelaskan bahwa sistem pengelolaan inventori manual menyebabkan proses pengendalian stok menjadi kurang efektif karena *admin* mengalami kesulitan dalam memantau perubahan persediaan secara cepat dan akurat. Penelitian tersebut juga menunjukkan bahwa penerapan sistem inventori berbasis *web* mampu membantu perusahaan dalam memonitor stok barang secara *real-time* sehingga proses pengambilan keputusan terkait pemesanan barang menjadi lebih cepat dan tepat.

Perkembangan teknologi informasi memungkinkan perusahaan menggunakan sistem inventori berbasis *web* yang dapat membantu proses pengendalian *Lead Time* secara lebih efektif. Sistem inventori modern mampu menampilkan kondisi stok secara *real-time*, mencatat riwayat transaksi barang, dan membantu *admin* mengetahui kapan stok barang mendekati batas *minimum*. Dengan adanya sistem tersebut, perusahaan dapat menentukan waktu pemesanan ulang dengan lebih akurat dan mengurangi risiko keterlambatan pengadaan barang.

Dalam konteks Toko Elegant Keramik, pengelolaan *Lead Time* menjadi sangat penting karena toko harus menyediakan berbagai jenis material bangunan dengan tingkat permintaan yang berbeda-beda. Jika proses pengelolaan stok masih dilakukan menggunakan Excel, maka *admin* membutuhkan waktu lebih lama untuk memeriksa kondisi persediaan barang sebelum melakukan pemesanan ulang. Kondisi tersebut dapat menyebabkan keterlambatan pengadaan barang dan menurunkan kualitas pelayanan *customer*.

Pada penelitian ini, sistem informasi material bangunan berbasis *web* dikembangkan untuk membantu *admin* memantau kondisi stok secara otomatis dan *real-time*. Sistem memungkinkan *admin* mengetahui jumlah persediaan barang yang tersedia sehingga proses pengambilan keputusan terkait pemesanan ulang dapat dilakukan lebih cepat dan efisien.

Dengan demikian, penerapan sistem inventori berbasis *web* diharapkan mampu membantu pengendalian *Lead Time* dan mengurangi risiko terjadinya kekurangan stok barang.

2.1.4 Stock Opname Sebagai Pengendalian Inventori

Stock opname merupakan proses pemeriksaan fisik persediaan secara berkala untuk mencocokkan data sistem dengan kondisi nyata di gudang. Kegiatan ini bertujuan untuk mengidentifikasi selisih stok, kesalahan pencatatan, maupun kehilangan barang [17].

Dalam kegiatan operasional perusahaan, selisih stok antara data sistem dan kondisi fisik barang dapat terjadi akibat berbagai faktor, seperti kesalahan pencatatan, keterlambatan pembaruan data, kehilangan barang, kerusakan produk, maupun kesalahan saat transaksi barang masuk dan barang keluar. Oleh karena itu, *stock opname* dilakukan untuk mengidentifikasi adanya perbedaan stok dan membantu perusahaan melakukan koreksi terhadap data inventori yang tidak sesuai.

Pada sistem inventori manual yang masih menggunakan Microsoft Excel, proses *stock opname* biasanya membutuhkan waktu yang cukup lama karena *admin* harus mencocokkan data barang satu per satu dengan kondisi fisik di gudang. Selain memakan waktu, metode manual juga meningkatkan risiko kesalahan perhitungan dan keterlambatan penyusunan laporan inventori. Jika jumlah barang yang dikelola sangat banyak, maka proses *stock opname* akan menjadi semakin kompleks dan sulit dilakukan secara efisien.

Penelitian [18] menjelaskan bahwa pencatatan inventori secara manual menyebabkan proses pelacakan pergerakan barang menjadi kurang efektif dan menghambat proses pengendalian stok. Penelitian tersebut menunjukkan bahwa penggunaan sistem inventori berbasis *web* mampu membantu perusahaan dalam melakukan *monitoring* stok dan mempercepat proses pencocokan data inventori dengan kondisi fisik barang.

Dalam sistem informasi inventori *modern*, proses *stock opname* dapat didukung oleh sistem digital yang terintegrasi dengan *database* inventori. Sistem memungkinkan *admin* melihat data stok secara *real-time*, riwayat transaksi barang, serta perubahan jumlah persediaan pada setiap *item* barang. Dengan demikian, proses pemeriksaan fisik barang menjadi lebih cepat, akurat, dan terstruktur dibandingkan metode pencatatan manual menggunakan *spreadsheet*.

Selain membantu pencocokan data stok, *stock opname* juga memiliki manfaat lain dalam pengendalian inventori, seperti mencegah kehilangan barang, meningkatkan pengawasan terhadap persediaan, serta membantu perusahaan mengevaluasi efektivitas sistem pengelolaan inventori yang digunakan. Hasil *stock opname* dapat digunakan sebagai

dasar untuk melakukan perbaikan terhadap prosedur pencatatan dan pengendalian stok agar lebih efektif dan efisien.

Dalam konteks Toko Elegant Keramik, kegiatan *stock opname* sangat penting karena toko memiliki banyak jenis material bangunan dengan jumlah stok yang cukup besar. Jika proses pencatatan masih dilakukan menggunakan Excel, maka *admin* akan mengalami kesulitan dalam mencocokkan data stok dengan kondisi barang di gudang. Kondisi tersebut dapat menyebabkan keterlambatan pelayanan *customer* dan menurunkan akurasi informasi inventori.

Pada penelitian ini, sistem informasi material bangunan berbasis *web* dikembangkan untuk membantu proses pengendalian inventori melalui pencatatan data stok yang terintegrasi dan *real-time*. Sistem memungkinkan setiap transaksi barang masuk dan barang keluar langsung memperbarui jumlah stok di *database* sehingga *admin* dapat lebih mudah melakukan proses *stock opname* dan *monitoring* inventori secara efisien.

2.1.5 Indikator Kinerja Manajemen Inventori

Keberhasilan manajemen inventori dapat diukur melalui beberapa indikator kinerja, antara lain tingkat akurasi stok, tingkat perputaran persediaan (*Inventory Turnover*), frekuensi kehabisan stok (*Stockout*), dan efisiensi biaya penyimpanan [19].

Salah satu indikator utama dalam manajemen inventori adalah tingkat akurasi stok (*Inventory Accuracy*). Indikator ini menunjukkan tingkat kesesuaian antara data stok yang tercatat di dalam sistem dengan jumlah stok fisik yang tersedia di gudang. Tingkat akurasi stok yang tinggi menunjukkan bahwa proses pencatatan dan pengelolaan inventori berjalan dengan baik, sedangkan tingkat akurasi yang rendah menunjukkan adanya permasalahan dalam sistem pengelolaan persediaan.

Indikator berikutnya adalah *Inventory Turnover* atau tingkat perputaran persediaan. *Inventory Turnover* digunakan untuk mengukur seberapa cepat barang persediaan terjual atau digunakan dalam periode tertentu. Semakin tinggi nilai *Inventory Turnover*, maka semakin cepat perputaran barang di gudang sehingga perusahaan dapat mengurangi risiko penumpukan stok dan biaya penyimpanan barang.

Selain itu, terdapat indikator *Stockout Frequency* atau frekuensi kehabisan stok. Indikator ini digunakan untuk mengukur seberapa sering perusahaan mengalami kondisi kekurangan barang sehingga tidak dapat memenuhi permintaan *customer*. Tingginya frekuensi *stockout* menunjukkan bahwa pengendalian persediaan belum berjalan dengan optimal dan dapat menyebabkan menurunnya kualitas pelayanan serta hilangnya peluang penjualan.

Indikator lainnya adalah efisiensi biaya penyimpanan (*Inventory Carrying Cost*). Biaya penyimpanan meliputi biaya gudang, biaya perawatan barang, biaya keamanan, dan risiko kerusakan barang. Pengelolaan inventori yang baik bertujuan untuk menjaga keseimbangan antara jumlah stok yang tersedia dengan biaya penyimpanan yang dikeluarkan perusahaan agar operasional tetap berjalan secara efisien.

Pada sistem inventori manual berbasis Excel, pengukuran indikator kinerja sering mengalami kendala karena data inventori tidak diperbarui secara otomatis dan proses rekapitulasi laporan membutuhkan waktu yang lebih lama. Kondisi tersebut menyebabkan *admin* kesulitan dalam memantau performa pengelolaan stok secara cepat dan akurat. Selain itu, keterlambatan pembaruan data juga dapat memengaruhi kualitas pengambilan keputusan terkait persediaan barang.

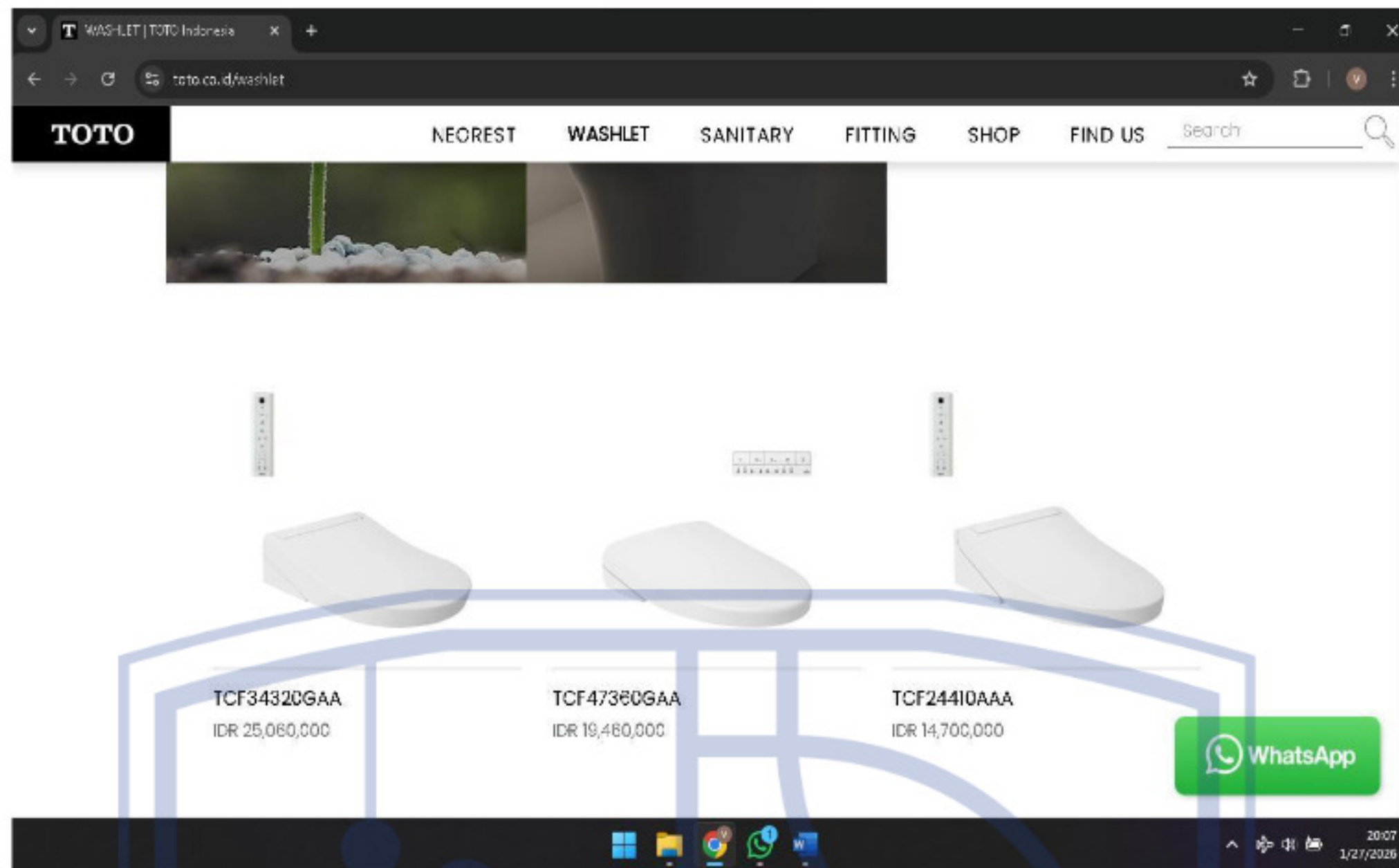
Penelitian [20] menunjukkan bahwa penerapan sistem inventori berbasis *web* mampu meningkatkan efisiensi pengelolaan stok, mempercepat akses data inventori, dan membantu perusahaan dalam memantau kondisi persediaan secara *real-time*. Sistem inventori digital juga mempermudah proses penyusunan laporan dan pengukuran indikator kinerja inventori secara lebih akurat.

Dalam penelitian ini, indikator kinerja manajemen inventori digunakan untuk menilai keberhasilan penerapan sistem informasi material bangunan berbasis web pada Toko Elegant Keramik. Sistem yang dikembangkan diharapkan mampu meningkatkan akurasi stok, mempercepat proses pengelolaan data inventori, mengurangi risiko *stockout*, serta meningkatkan efisiensi operasional toko. Selain itu, sistem juga diharapkan mampu meningkatkan kualitas pelayanan *customer* melalui penyediaan informasi produk dan stok secara *real-time*.

2.2 Sistem Informasi Berbasis Web

Sistem informasi berbasis *web* adalah sistem digital yang berjalan menggunakan teknologi internet dan dapat diakses melalui *browser* tanpa memerlukan instalasi khusus. Sistem berbasis *web* menawarkan fleksibilitas, kemampuan akses lintas perangkat, serta kemudahan pemeliharaan dibandingkan sistem *desktop* [21][22]. Selain itu, aplikasi *web* memungkinkan integrasi data secara *real-time* dan responsif terhadap kebutuhan bisnis [22].

Salah satu contoh nyata implementasi sistem informasi berbasis *website* dapat ditemukan pada platform TOTO Indonesia. *Website* ini menyediakan informasi harga dan ketersediaan produk sanitaris



Gambar 2.1 Website TOTO Indonesia ([TOTO Indonesia](https://www.toto.co.id))

Website TOTO Indonesia menerapkan sistem informasi berbasis *web* yang memungkinkan *customer* mengakses informasi produk dan ketersediaan stok secara mandiri. Fitur utama yang relevan dengan penelitian ini meliputi:

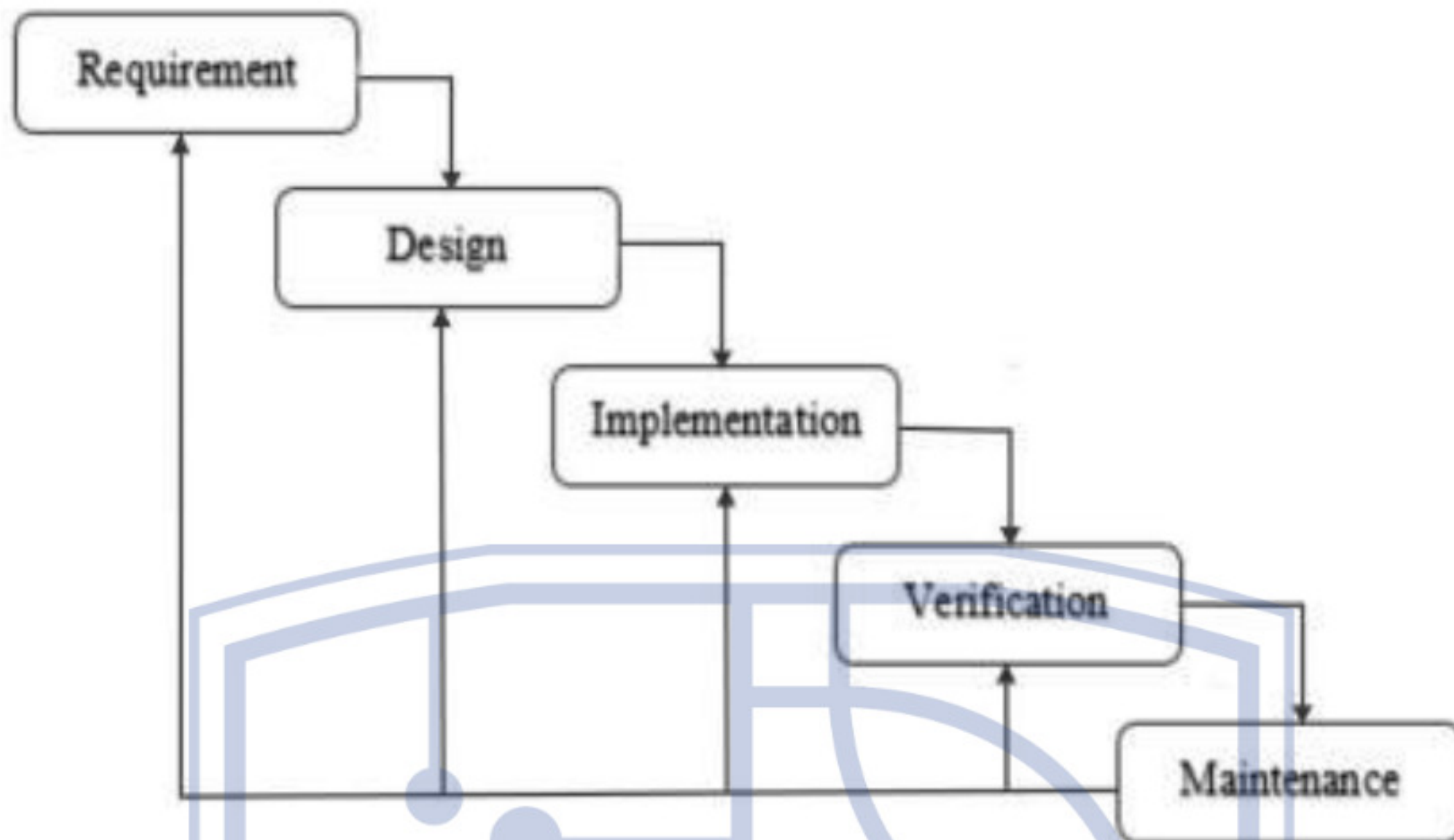
1. Katalog produk dengan kategori,
2. Informasi harga dan spesifikasi,
3. Pencarian produk.

Namun, sistem pada penelitian ini akan difokuskan pada pengelolaan inventori dari sisi *admin*, bukan hanya *display* produk.

Metode *Waterfall* merupakan model pengembangan perangkat lunak yang bersifat sekuensial dan linear, di mana setiap tahap harus diselesaikan terlebih dahulu sebelum melanjutkan ke tahap berikutnya. Model ini cocok digunakan pada pengembangan sistem yang kebutuhan fungsionalnya sudah jelas sejak awal dan relatif tidak mengalami perubahan selama proses pengembangan [23][24].

Pada penelitian pengembangan sistem informasi inventori berbasis *web* di Toko Elegant Keramik, metode *Waterfall* dipilih karena kebutuhan sistem telah terdefinisi dengan baik, alur proses bisnis relatif stabil, serta sesuai untuk pengembangan sistem berskala kecil hingga menengah.

Tahapan Metode *Waterfall*:



Gambar 2.2 Metode *Waterfall* [25]

1. *Requirement* (Analisis Kebutuhan)

Pada tahap ini, pengembang sistem melakukan proses komunikasi intensif dengan pengguna untuk mengidentifikasi kebutuhan perangkat lunak yang akan dibangun, termasuk batasan-batasan yang mungkin ada. Informasi dikumpulkan melalui berbagai metode seperti wawancara, observasi langsung, diskusi kelompok, atau kuesioner. Seluruh data yang terkumpul kemudian dianalisis guna merumuskan spesifikasi kebutuhan sistem secara jelas dan terstruktur.

2. *Design* (Perancangan Sistem)

Setelah kebutuhan sistem ditentukan, tahap berikutnya adalah menyusun desain teknis dari sistem. Proses ini mencakup perancangan struktur perangkat lunak, spesifikasi kebutuhan perangkat keras, serta perencanaan arsitektur sistem secara menyeluruh. Tujuannya adalah untuk menciptakan kerangka kerja yang menjadi acuan utama dalam proses pengembangan aplikasi.

3. *Implementation* (Implementasi)

Pada tahap implementasi, pengembang mulai menerjemahkan desain sistem ke dalam kode program. Pengembangan dilakukan secara modular, dimulai dari unit-unit kecil (modul) yang kemudian akan digabungkan. Masing-masing unit diuji terlebih dahulu dalam proses yang dikenal sebagai *unit testing* untuk memastikan bahwa setiap fungsi berjalan sesuai dengan spesifikasi.

4. *Verification* (Pengujian Sistem)

Tahap ini merupakan proses pengujian sistem secara keseluruhan untuk memastikan bahwa aplikasi memenuhi kebutuhan dan spesifikasi yang telah ditentukan. Verifikasi meliputi berbagai jenis pengujian seperti *unit testing* pada komponen individu, *Integration Testing* untuk menguji interaksi antar modul, serta *User Acceptance Testing* (UAT) yang melibatkan pengguna akhir untuk menilai apakah sistem sudah sesuai dengan harapan mereka.

5. *Maintenance* (Pemeliharaan Sistem)

Setelah sistem diimplementasikan dan digunakan, dilakukan tahap pemeliharaan untuk memastikan perangkat lunak tetap berjalan dengan baik. Pemeliharaan meliputi perbaikan *bug* yang tidak terdeteksi saat pengujian, pembaruan fitur, serta penyesuaian terhadap perubahan kebutuhan pengguna atau lingkungan operasional.

Gambar 2.2 di atas menggambarkan alur kerja metode *Waterfall* secara umum. Pada penelitian ini, model *Waterfall* tidak diterapkan secara generik, melainkan diadaptasi secara spesifik sesuai kebutuhan pengembangan Sistem Informasi Inventori Material Bangunan Berbasis *web* pada Toko Elegant Keramik. Setiap tahap memiliki aktivitas konkret yang berlandaskan kondisi nyata di lapangan.

Tabel 2.1 berikut memetakan setiap tahap *Waterfall* dengan aktivitas spesifik yang dilakukan pada penelitian ini. Pada tahap *Requirements Analysis*, dilakukan observasi langsung dan wawancara dengan pemilik Toko Elegant Keramik untuk mengidentifikasi permasalahan pencatatan inventori manual. Pada tahap Sistem *Design*, dirancang arsitektur sistem berbasis *web*, ERD, *Use Case Diagram*, *Activity Diagram*, dan *Wireframe* antarmuka *admin* maupun *customer*.

Pada tahap *Implementation*, sistem dibangun menggunakan ReactJS 19.2.0 sebagai *front end* dan Cloud Firestore sebagai *database real-time*. Pada tahap *Testing*, dilakukan pengujian fungsional menggunakan metode *Black Box Testing* untuk memverifikasi seluruh fitur sistem. Terakhir, pada tahap *Deployment & Maintenance*, sistem di-*deploy* pada *server* lokal dan dilakukan pemeliharaan berkala untuk memastikan sistem berjalan optimal.

Tabel 2.1 Pemetaan Tahap *Waterfall* Dengan Aktivitas Penelitian

No	Tahap	Aktivitas	Output
1	<i>Requirements Analysis</i> (Analisis Kebutuhan)	Observasi langsung di Toko Elegant Keramik, wawancara dengan pemilik dan <i>admin</i> toko, identifikasi kebutuhan fungsional (kelola stok, informasi produk untuk <i>customer</i>) dan non-fungsional (performa, keamanan data)	Dokumen spesifikasi kebutuhan sistem, daftar kebutuhan fungsional dan non-fungsional
2	<i>System Design</i> (Perancangan Sistem)	Perancangan arsitektur sistem (<i>front end</i> ReactJS – Firestore), pembuatan <i>Use Case Diagram</i> , <i>Activity Diagram</i> , dan Flowchart, perancangan struktur <i>Database</i> di Firestore, perancangan <i>Mockup</i> antarmuka halaman <i>admin</i> dan <i>customer</i>	Diagram arsitektur sistem, <i>Use Case Diagram</i> , <i>Activity Diagram</i> , desain struktur koleksi <i>Firestore</i> , <i>Wireframe/Mockup</i> antarmuka
3	<i>Implementation</i> (Implementasi)	Pengkodean <i>front end</i> menggunakan ReactJS 19.2.0 dengan React Router dan Axios, konfigurasi Firebase (<i>Firestore Database</i> dan <i>Firebase Authentication</i>), pengembangan fitur kelola barang, pencarian produk, dan <i>update</i> stok oleh <i>admin</i>	Aplikasi <i>web</i> yang dapat dijalankan secara lokal, seluruh modul fungsional telah selesai dikembangkan
4	<i>Testing</i> (Pengujian Sistem)	Pengujian menggunakan metode <i>Black Box Testing</i> terhadap seluruh fitur utama (kelola data barang, pencarian produk, <i>update</i> stok, <i>login admin</i>), pengujian dilakukan oleh tim pengembang dan pihak toko	Laporan hasil pengujian <i>Black Box Testing</i> , daftar bug yang ditemukan dan diperbaiki
5	<i>Deployment & Maintenance</i> (Pemeliharaan)	Sistem di- <i>deploy</i> pada <i>server</i> lokal di Toko Elegant Keramik, pelatihan singkat kepada <i>admin</i> toko, perbaikan <i>bug</i> pasca-pengujian, pemantauan performa sistem selama periode awal penggunaan	Sistem berjalan stabil di lingkungan produksi, dokumentasi penggunaan sistem untuk <i>admin</i>

Tabel 2.2 Tabel Perbandingan *Waterfall*, *Agile*, *Prototyping*

Aspek	Waterfall	Agile	Prototyping
Pendekatan	Linear dan berurutan	Iteratif dan <i>incremental</i>	Berbasis pembuatan <i>prototype</i>
Perubahan Kebutuhan	Sulit diubah setelah tahap berjalan	Sangat fleksibel terhadap perubahan	Fleksibel di awal, tergantung revisi <i>prototype</i>
Keterlibatan <i>User</i>	Terbatas di awal dan akhir	Sangat tinggi (berkelanjutan)	Tinggi pada tahap evaluasi <i>prototype</i>
Dokumentasi	Lengkap dan terstruktur	Lebih sedikit, fokus pada <i>development</i>	Cukup, tergantung kebutuhan
Kesesuaian Proyek	Kebutuhan jelas dan stabil	Kebutuhan sering berubah	Kebutuhan belum jelas
Keunggulan	Terstruktur, mudah dikontrol	Adaptif dan cepat	Memudahkan visualisasi sistem
Kelemahan	Kurang fleksibel terhadap perubahan	Sulit diprediksi <i>timeline</i>	Bisa memakan waktu tambahan

Berdasarkan Tabel 2.2, metode *Waterfall* dipilih dalam penelitian ini karena memiliki karakteristik yang paling sesuai dengan kondisi dan kebutuhan pengembangan Sistem Informasi Inventori pada Toko Elegant Keramik. Dari aspek pendekatan, *Waterfall* menggunakan tahapan yang terstruktur dan berurutan sehingga memudahkan proses pengembangan sistem. Dari aspek dokumentasi, metode ini menghasilkan dokumentasi yang lengkap dan terorganisir sehingga mempermudah proses pengelolaan proyek serta pemeliharaan sistem di masa mendatang.

Selain itu, kebutuhan sistem pada penelitian ini telah teridentifikasi dengan cukup jelas melalui kegiatan observasi dan wawancara sehingga perubahan kebutuhan selama proses pengembangan diperkirakan relatif kecil. Kondisi tersebut sesuai dengan karakteristik *Waterfall* yang lebih efektif digunakan pada proyek dengan kebutuhan yang stabil. Dibandingkan metode *Agile* yang memerlukan keterlibatan pengguna secara intensif dan perubahan kebutuhan yang berkelanjutan, maupun metode *Prototyping* yang lebih sesuai untuk kebutuhan yang masih belum terdefinisi secara jelas, *Waterfall* dinilai lebih tepat untuk diterapkan pada penelitian ini.

Meskipun *Waterfall* memiliki keterbatasan dalam mengakomodasi perubahan kebutuhan pada tahap pengembangan yang sedang berjalan, risiko tersebut dapat

diminimalkan melalui analisis kebutuhan yang dilakukan secara menyeluruh pada tahap awal. Oleh karena itu, metode Waterfall dianggap mampu memberikan proses pengembangan yang sistematis, terdokumentasi dengan baik, serta memudahkan pengendalian dan evaluasi proyek.

2.3 Teknologi ReactJS dan Firestore

Pada penelitian ini, pengembangan sistem informasi inventori berbasis *web* menggunakan dua teknologi utama, yaitu ReactJS sebagai *framework front end* dan Cloud Firestore dari Firebase sebagai layanan *back end* dan basis data. Pemilihan kedua teknologi ini didasarkan pada pertimbangan kebutuhan sistem yang memerlukan antarmuka pengguna yang responsif, pengelolaan data secara *real-time*, serta kemudahan integrasi tanpa memerlukan pengelolaan *server* secara mandiri [26].

Berikut adalah perbandingan beberapa teknologi *front end* yang umum digunakan:

Tabel 2.3 Perbandingan ReactJS, Vue, dan Angular

Aspek	ReactJS	Vue.js	Angular
Jenis	<i>Library</i>	<i>Framework</i>	<i>Full Framework</i>
Performa	Tinggi (<i>Virtual DOM</i>)	Tinggi	Cukup
Kemudahan Belajar	Sedang	Mudah	Cukup sulit
<i>Reusability</i>	Sangat tinggi	Tinggi	Tinggi
Kompleksitas	Ringan	Ringan	Kompleks

Berdasarkan Tabel 2.3, ReactJS dipilih karena memiliki performa tinggi melalui penggunaan *Virtual DOM*, tingkat *reusability* yang sangat tinggi, serta kompleksitas yang relatif ringan. Karakteristik tersebut sesuai dengan kebutuhan Sistem Informasi Inventori yang memerlukan tampilan responsif dan pengelolaan komponen antarmuka yang dapat digunakan kembali.

Selain itu, ReactJS didukung oleh ekosistem yang luas dan dokumentasi yang lengkap sehingga memudahkan proses pengembangan. Dibandingkan Vue.js, ReactJS menawarkan fleksibilitas yang lebih tinggi, sedangkan Angular dinilai kurang sesuai karena

memiliki kompleksitas yang lebih besar untuk kebutuhan sistem pada penelitian ini. Selain *front end*, pemilihan *Database* juga penting. Berikut perbandingannya:

Tabel 2.4 Perbandingan Firestore, MySQL, PostgreSQL, dan MongoDB

Aspek	Firestore	MySQL	PostgreSQL	MongoDB
Tipe	NoSQL	Relasional	Relasional	NoSQL
<i>Real-time</i>	Ada	Tidak	Tidak	Tidak langsung
<i>Setup</i>	Tidak perlu <i>server</i>	Perlu <i>server</i>	Perlu <i>server</i>	Perlu <i>server</i>
Kompleksitas	Rendah	Sedang	Tinggi	Sedang

Berdasarkan Tabel 2.4, Firestore dipilih karena lebih praktis dan sesuai dengan kebutuhan sistem. Firestore sudah mendukung *real-time update*, jadi ketika stok barang berubah, data bisa langsung *update* tanpa perlu *refresh*. Selain itu, Firestore juga bersifat *serverless*, jadi tidak perlu *setup back end* atau *server* sendiri. Ini membuat proses pengembangan jadi lebih cepat dan lebih simpel, terutama untuk skala UMKM.

Sementara itu, MySQL dan PostgreSQL tidak dipilih karena perlu konfigurasi *server* dan tidak mendukung *real-time* secara langsung. MongoDB sebenarnya cukup fleksibel, tetapi tetap membutuhkan *setup back end* tambahan dibanding Firestore.

Berdasarkan perbandingan yang sudah dilakukan, alasan utama pemilihan teknologi dalam penelitian ini adalah:

1. ReactJS memudahkan pembuatan tampilan yang bisa digunakan berulang (*reusable component*), cocok untuk menampilkan banyak produk
2. Virtual DOM membuat performa aplikasi lebih cepat saat terjadi perubahan data
3. Firestore mendukung *update* data secara *real-time*, sehingga informasi stok selalu *up-to-date*
4. Tidak perlu *setup server*, sehingga lebih praktis dan hemat biaya
5. Cocok untuk sistem skala kecil hingga menengah seperti UMKM

Dengan kombinasi ReactJS dan Firestore, sistem yang dibangun menjadi lebih responsif, efisien, dan mudah dikembangkan.

2.3.1 ReactJS

ReactJS (React) adalah *library* JavaScript sumber terbuka yang dikembangkan oleh Meta (*Facebook*) dan dirilis pada tahun 2013. ReactJS digunakan untuk membangun

antarmuka pengguna (*user interface*) yang bersifat deklaratif dan berbasis komponen, sehingga memungkinkan pengembang membangun antarmuka yang kompleks dari potongan-potongan kecil yang dapat digunakan kembali (*reusable components*) [26]. ReactJS menggunakan konsep *Virtual DOM* (representasi virtual dari DOM yang disimpan di memori) yang disinkronkan dengan DOM nyata melalui proses *reconciliation*, sehingga hanya bagian yang berubah yang diperbarui dan performa *rendering* menjadi lebih optimal [26].

Dalam penelitian ini, ReactJS versi 19.2.0 digunakan bersama *library* React Router untuk navigasi antar halaman dan Axios untuk komunikasi dengan Firebase. Pendekatan berbasis komponen pada ReactJS sangat sesuai untuk menampilkan daftar produk inventori yang berjumlah banyak secara efisien karena setiap kartu produk dapat dibangun sebagai komponen yang dapat digunakan ulang [26].

2.3.2 Firestore

Firebase adalah platform pengembangan aplikasi berbasis *cloud* yang dikembangkan oleh *Google*. Salah satu layanan utama Firebase yang digunakan dalam penelitian ini adalah Firestore, yaitu basis data NoSQL berbasis dokumen yang mendukung sinkronisasi data secara *real-time*. Firestore menyimpan data dalam format koleksi dan dokumen (bukan tabel relasional), sehingga fleksibel untuk menyimpan data inventori dengan struktur yang beragam.

Selain itu, Firebase menyediakan layanan autentikasi pengguna (*Firebase Authentication*) dan *hosting* yang dapat diintegrasikan langsung dengan aplikasi ReactJS. Kelebihan utama Firebase dibanding basis data konvensional adalah kemudahan konfigurasi, tidak memerlukan *server* terpisah, dan biaya operasional yang minimal untuk skala UMKM [24][27].

2.4 System Usability Scale (SUS)

System Usability Scale (SUS) merupakan metode evaluasi *usability* yang dikembangkan oleh John Brooke pada tahun 1986 untuk mengukur tingkat kemudahan penggunaan suatu sistem berdasarkan persepsi pengguna[28]. Metode ini banyak digunakan karena sederhana, cepat, mudah diterapkan, serta mampu menghasilkan pengukuran yang reliabel terhadap *usability* berbagai jenis sistem, seperti aplikasi *web*, *desktop*, maupun *mobile* [28].

SUS terdiri dari 10 pernyataan yang dijawab menggunakan skala Likert 1–5, mulai dari Sangat Tidak Setuju (STS) hingga Sangat Setuju (SS). Pernyataan bernomor ganjil merupakan pernyataan positif, sedangkan pernyataan bernomor genap merupakan pernyataan negatif. Penyusunan positif-negatif bertujuan mengurangi bias jawaban responden [28].

Tabel 2.5 Pernyataan SUS

No	Pernyataan SUS
1	Saya berpikir akan menggunakan sistem ini lagi.
2	Saya merasa sistem ini terlalu rumit digunakan.
3	Saya merasa sistem ini mudah digunakan.
4	Saya membutuhkan bantuan orang lain untuk menggunakan sistem ini.
5	Saya merasa fitur-fitur sistem berjalan sebagaimana mestinya.
6	Saya merasa terdapat banyak inkonsistensi pada sistem ini.
7	Saya merasa kebanyakan orang dapat mempelajari sistem ini dengan cepat.
8	Saya merasa sistem ini membingungkan.
9	Saya merasa percaya diri saat menggunakan sistem ini.
10	Saya perlu mempelajari banyak hal sebelum menggunakan sistem ini.

Berdasarkan Tabel 2.5, kuesioner *System Usability Scale* (SUS) terdiri atas 10 pernyataan yang digunakan untuk mengukur tingkat kemudahan penggunaan sistem berdasarkan persepsi pengguna. Pernyataan bernomor ganjil (1, 3, 5, 7, dan 9) merupakan pernyataan positif, sedangkan pernyataan bernomor genap (2, 4, 6, 8, dan 10) merupakan pernyataan negatif. Setiap pernyataan dijawab menggunakan skala Likert 1–5, kemudian hasil jawaban responden akan diolah menggunakan aturan perhitungan SUS untuk memperoleh skor *usability* sistem [28].

2.4.1 Perhitungan Skor SUS

Perhitungan skor SUS dilakukan melalui proses konversi nilai jawaban responden. Konversi diperlukan karena instrumen SUS menggunakan kombinasi pernyataan positif dan negatif sehingga seluruh skor harus memiliki arah penilaian yang sama [28].

Langkah-langkah perhitungan:

1. Pernyataan ganjil (1,3,5,7,9): $\text{Skor} = X - 1$.
2. Pernyataan genap (2,4,6,8,10): $\text{Skor} = 5 - X$.
3. Jumlahkan seluruh skor hasil konversi sehingga diperoleh nilai 0–40.
4. Kalikan total skor dengan 2,5 sehingga diperoleh skor SUS pada rentang 0–100 [28].

Pengali 2,5 digunakan untuk mengonversi skor maksimum 40 menjadi skala standar 100 sehingga hasil penelitian dapat dibandingkan dengan penelitian lain yang menggunakan SUS [28].

Setelah skor masing-masing responden diperoleh, dihitung rata-rata seluruh responden menggunakan rumus:

$$\bar{X} = \Sigma X / n$$

dengan $\bar{X}$ = rata-rata skor SUS, ΣX = jumlah seluruh skor SUS responden, dan n = jumlah responden. Nilai rata-rata inilah yang digunakan sebagai skor *usability* sistem secara keseluruhan [29], [30].

Contoh Perhitungan

Misalnya seorang responden memberikan jawaban:

Q1=4, Q2=2, Q3=5, Q4=2, Q5=4, Q6=1, Q7=5, Q8=2, Q9=4, Q10=1.

Setelah dikonversi sesuai aturan SUS diperoleh total skor konversi sebesar 34. Skor SUS responden tersebut adalah:

$$34 \times 2,5 = 85$$

Perhitungan dilakukan untuk seluruh responden, kemudian dicari nilai rata-ratanya sebagai skor *usability* sistem.

2.4.2 Interpretasi Skor SUS

Setelah diperoleh skor SUS dari setiap responden, langkah selanjutnya adalah menginterpretasikan skor tersebut untuk mengetahui tingkat *usability* sistem. Interpretasi skor SUS dilakukan berdasarkan kategori *Grade*, *Adjective Rating*, dan *Acceptability* sehingga dapat memberikan gambaran mengenai tingkat penerimaan sistem oleh pengguna [29].

Tabel 2.6 interpretasi Skor SUS

Skor	Grade	Adjective	Acceptability
>80.3	A	<i>Excellent</i>	<i>Acceptable</i>
68-80.3	B	<i>Good</i>	<i>Acceptable</i>
68	C	<i>OK</i>	<i>Marginal</i>
51-67	D	<i>Poor</i>	<i>Marginal</i>
<51	F	<i>Awful</i>	<i>Not Acceptable</i>

Berdasarkan Tabel 2.4, skor SUS yang semakin tinggi menunjukkan tingkat *usability* sistem yang semakin baik. Secara umum, sistem dinilai memiliki *usability* yang baik apabila memperoleh skor 68 atau lebih, sedangkan skor di bawah 68 menunjukkan bahwa sistem masih memerlukan perbaikan. Selain itu, kategori *Grade*, *Adjective Rating*, dan

Acceptability digunakan untuk mempermudah interpretasi terhadap tingkat kualitas dan penerimaan sistem oleh pengguna [29], [30].

2.5 Penelitian Terdahulu

Penelitian terdahulu merupakan kajian terhadap penelitian-penelitian yang telah dilakukan sebelumnya yang memiliki keterkaitan dengan topik penelitian ini. Tujuan dari kajian ini adalah untuk mengetahui posisi penelitian yang dilakukan, mengidentifikasi kelebihan dan kekurangan penelitian sebelumnya, serta menemukan celah penelitian (*Research Gap*) yang dapat dikembangkan lebih lanjut.

Berikut beberapa penelitian terdahulu yang relevan dengan sistem informasi inventori berbasis *web*:

Tabel 2.7 Peneliti Terdahulu

No	Peneliti (Tahun)	Judul	Metode	Hasil	Keterbatasan
1	Djoni & Tony (2024)	Sistem Manajemen Stok Barang Berbasis <i>web</i> pada Toko Bangunan	<i>Web-based system</i>	Meningkatkan efisiensi pengelolaan stok dan pencatatan	Tidak menyediakan akses informasi untuk <i>customer</i>
2	Farhan & Ghani (2023)	Sistem Informasi Inventori pada Perusahaan Manufaktur	<i>Waterfall</i>	Meningkatkan akurasi dan <i>monitoring</i> stok <i>real-time</i>	Fokus hanya pada internal perusahaan
3	Edison dkk. (2024)	Sistem Inventori dengan Metode ROP dan <i>Safety Stock</i>	Perhitungan inventori + <i>Web</i>	Memberikan notifikasi stok <i>minimum</i>	Kurang fokus pada <i>user Interface</i> dan kemudahan pengguna
4	Schiller dkk. (2023)	Aplikasi Pengelolaan Stok Barang pada Toko <i>Retail</i>	<i>Web-based system</i>	Mempermudah pencarian dan pengelolaan data barang	Tidak terintegrasi dengan sistem <i>customer</i>
5	Hikmal dkk. (2025)	Sistem Inventori Berbasis <i>Web</i> dengan	<i>Black Box Testing</i>	Sistem berjalan sesuai fungsi dan meningkatkan akurasi data	Tidak membahas pengembangan

		Pengujian Fungsional			fitur interaktif <i>customer</i>
--	--	----------------------	--	--	----------------------------------

Berdasarkan Tabel 2.5, dapat disimpulkan bahwa penelitian-penelitian terdahulu umumnya berfokus pada pengembangan sistem informasi inventori berbasis *web* yang bertujuan untuk meningkatkan efisiensi pengelolaan stok, akurasi pencatatan data, serta kemudahan dalam melakukan *monitoring* persediaan secara *real-time*. Meskipun demikian, sebagian besar penelitian tersebut masih memiliki keterbatasan, terutama dalam hal ruang lingkup yang hanya mencakup kebutuhan internal organisasi.

Selain itu, aspek interaksi dengan *customer*, seperti penyediaan akses informasi produk secara mandiri, belum menjadi fokus utama. Beberapa penelitian juga menunjukkan kelemahan pada sisi antarmuka pengguna (*User Interface*) serta kemudahan penggunaan sistem (*Usability*), sehingga belum sepenuhnya optimal dalam mendukung pengalaman pengguna secara menyeluruh.

Oleh karena itu, penelitian ini dilakukan untuk mengisi kesenjangan penelitian (*Research Gap*) yang telah diidentifikasi, dengan mengembangkan sistem informasi inventori berbasis *web* yang tidak hanya mendukung pengelolaan data stok secara internal, tetapi juga mengintegrasikan fitur akses informasi produk bagi *customer*.

Sistem yang diusulkan diharapkan mampu meningkatkan efisiensi pengelolaan stok, menjaga akurasi data, serta memperbaiki kualitas layanan melalui penyediaan informasi yang cepat, tepat, dan mudah diakses. Dengan demikian, penelitian ini memberikan kontribusi dalam pengembangan sistem inventori yang lebih komprehensif, baik dari sisi pengelolaan internal maupun interaksi eksternal dengan *customer*.

2.6 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk memastikan sistem berjalan sesuai kebutuhan pengguna dan bebas dari kesalahan fungsional. Salah satu metode yang umum digunakan adalah *Black Box Testing*, yaitu pengujian yang berfokus pada fungsi sistem tanpa melihat struktur internal program. Metode ini efektif digunakan pada tahap akhir untuk memastikan perangkat lunak memenuhi spesifikasi yang telah ditentukan [22][31].

Black Box Testing adalah metode pengujian perangkat lunak yang berfokus pada pengujian fungsionalitas sistem tanpa memperhatikan struktur internal program atau kode sumber. Pengujian ini dilakukan dengan cara memberikan *input* tertentu ke dalam sistem dan kemudian mengamati apakah *output* yang dihasilkan telah sesuai dengan hasil yang diharapkan. Metode ini menilai sistem dari sudut pandang pengguna (*user-oriented*),

sehingga sangat relevan untuk menguji aplikasi berbasis *web* yang digunakan oleh *admin* maupun *customer* [32].

1. Konsep *Black Box Testing*

Dalam *Black Box Testing*, penguji tidak perlu mengetahui bagaimana sistem dibangun secara teknis, melainkan hanya memeriksa apakah fungsi-fungsi utama sistem dapat berjalan dengan benar. Fokus utama pengujian meliputi:

- a. Validasi *input* dan *output*
- b. Fungsi sistem sesuai dengan kebutuhan pengguna
- c. Penanganan kesalahan (*error handling*)
- d. Respons sistem terhadap *input* yang tidak *valid*

Metode ini sangat cocok digunakan pada tahap akhir pengembangan sistem karena mampu memastikan bahwa seluruh fitur telah berjalan sesuai dengan spesifikasi yang telah dirancang sebelumnya [23].

2. Teknik dalam *Black Box Testing*



Gambar 2.3 *Black Box Testing* [33]

Beberapa teknik yang umum digunakan dalam *Black Box Testing* antara lain:

- a. *Equivalence Partitioning*, yaitu membagi data *input* ke dalam kelompok *valid* dan tidak *valid* [23].
- b. *Boundary Value Analysis*, yaitu pengujian pada batas nilai *minimum* dan *maksimum input* [23][32].
- c. *Error Guessing*, yaitu pengujian berdasarkan kemungkinan kesalahan yang sering terjadi [32].
- d. *Functional Testing*, yaitu pengujian setiap fungsi utama sistem [23]

Menurut penelitian [34] pengujian fungsional merupakan bagian dari *Black Box Testing* yang dilakukan untuk memverifikasi bahwa perangkat lunak telah memenuhi spesifikasi kebutuhan tanpa memperhatikan struktur internal kode program. Selain itu,

penelitian [35] menjelaskan bahwa *Functional Testing* berfokus pada validasi *input* dan *output* sistem untuk memastikan kesesuaian dengan *Requirement* yang telah ditentukan.

Penggunaan pendekatan ini juga didukung oleh penelitian terdahulu. Penelitian [3] dalam pengembangan sistem manajemen stok barang berbasis *web* menerapkan pengujian fungsional untuk memastikan fitur pengelolaan stok dan pelaporan berjalan sesuai kebutuhan operasional. Demikian pula, penelitian [36] menyatakan bahwa pengujian difokuskan pada fungsi utama sistem guna menjamin keakuratan pencatatan serta penyajian informasi stok.

Berdasarkan landasan teori dan penelitian terdahulu tersebut, dapat disimpulkan bahwa penggunaan teknik *Functional Testing* dalam penelitian ini relevan dan sesuai, karena sistem yang dikembangkan lebih menitikberatkan pada pengujian fungsi operasional dibandingkan dengan pengujian struktur internal atau performa sistem secara mendalam.

2.7 Profil Mitra Penelitian (Toko Elegant Keramik)

Toko Elegant Keramik merupakan usaha dagang yang bergerak di bidang penjualan material bangunan, khususnya keramik, sanitaris, plafon, dan produk pendukung lainnya. Toko ini sudah berjalan selama 2 tahun berlokasi di Jl. Platina Raya No.101, Titi Papan, Kecamatan Medan Deli, Kota Medan, Sumatera Utara.

Dalam kegiatan operasional sehari-hari, pengelolaan stok barang di Toko Elegant Keramik masih dilakukan secara semi-manual (Microsoft Excel dan Operator) menggunakan pencatatan sederhana. Kondisi tersebut berpotensi menimbulkan ketidaksesuaian data stok, keterlambatan pencarian informasi barang, serta kesulitan dalam penyusunan laporan inventori.

Oleh karena itu, penelitian ini dilakukan dengan menjadikan Toko Elegant Keramik sebagai mitra penelitian untuk mengembangkan Sistem Informasi Inventori Berbasis *web* yang mampu membantu pengelolaan stok secara lebih terstruktur, akurat, dan efisien. Dokumentasi berupa foto toko dan aktivitas operasional disertakan pada bagian lampiran sebagai bukti pelaksanaan penelitian.



Gambar 2.4 Informasi Mitra

Elegant Keramik adalah sebuah toko yang sudah berjalan selama kurang lebih 2 tahun. Toko ini beralamat di Jl. Platina Raya No.101, Titi Papan, Kecamatan Medan Deli, Kota Medan, Sumatera Utara 20245. Dalam menjalankan operasional sehari-harinya, Elegant Keramik dibantu oleh 3 orang karyawan. Jika ingin menghubungi pihak toko untuk menanyakan informasi atau melakukan pemesanan, dapat menghubungi nomor 081262432853.