**LAPORAN FINAL PROJECT**

**MATA KULIAH ALGORITMA & PEMROGRAMAN**

**SEMESTER GENAP 2024/2025**

**SISTEM KASIR MINIMARKET**



**Dosen Pengampu:**

**Retno Mumpuni, S.Kom, M.Sc.**

**Disusun Oleh:**

**Raja Samudra Tandepadang (24081010063)**

**Tryana Bagus Saputra (24081010086)**

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAWA TIMUR**

**2025**

1. **Deskripsi Program**
2. **Penjelasan Program**
   1. **Inisialisasi Library**
      1. #include <stdio.h>
      2. #include <time.h>
      3. #define MAX 20
      4. struct produk {
      5. char nama[25];
      6. int harga;
      7. int stok;
      8. int pesan;
      9. int terjual;
      10. };
   2. **Fungsi**
      1. void listProduk(struct produk prod[]) {
      2. struct produk data[] = {
      3. //NAMA                 HARGA   STOK   PESAN   TERJUAL
      4. {"Beras 5kg",          68000,   30,     0,       0},
      5. {"Minyak Goreng 1L",   18500,   25,     0,       0},
      6. {"Gula Pasir 1kg",     16000,   35,     0,       0},
      7. {"Garam 500gr",        4000,    40,     0,       0},
      8. {"Tepung Terigu 1kg",  12000,   30,     0,       0},
      9. {"Telur 1/2 lusin",    12000,   50,     0,       0},
      10. {"Kecap Manis",        10000,   20,     0,       0},
      11. {"Saus Sambal",        9000,    18,     0,       0},
      12. {"Mie Instan",         3000,    60,     0,       0},
      13. {"Kopi Sachet",        2500,    90,     0,       0},
      14. {"Teh Celup Isi 30",   9000,    55,     0,       0},
      15. {"Air Mineral 600ml",  3500,    100,    0,       0},
      16. {"Air Mineral 1.5L",   6000,    90,     0,       0},
      17. {"Susu UHT 1L",        17000,   40,     0,       0},
      18. {"Susu Kental Manis",  12000,   25,     0,       0},
      19. {"Pasta Gigi",         9000,    35,     0,       0},
      20. {"Sikat Gigi",         6000,    40,     0,       0},
      21. {"Shampoo",            16000,   30,     0,       0},
      22. {"Sabun Cuci Piring",  13000,   45,     0,       0},
      23. {"Sabun Mandi",        3500,    55,     0,       0}
      24. };
      26. for (int i = 0; i < MAX; i++) {
      27. prod[i] = data[i];
      28. }
      29. }
   3. **Fungsi**
      1. void tampilkanMenu(int \*menu) {
      2. printf("\nSELAMAT DATANG DI DUROMART\n\n");
      3. printf("DAFTAR MENU\n");
      4. printf("1. Pesan\n");
      5. printf("2. Cek Stok Produk\n");
      6. printf("3. Rekap Penjualan\n");
      7. while (\*menu < 1 || \*menu > 3) {
      8. printf("Pilih menu: ");
      9. scanf("%d", menu);
      10. if (\*menu < 1 || \*menu > 3) {
      11. printf("Input tidak valid, silahkan coba lagi.\n");
      12. }
      13. }
      14. }
   4. **Fungsi**
      1. void cekStok(struct produk prod[]) {
      2. printf("\nSTOK PRODUK\n\n");
      3. printf("KODE\tNAMA PRODUK\t\tSTOK\tKODE\tNAMA PRODUK\t\tSTOK\n");
      4. for (int i = 0; i < MAX / 2; i++) {
      5. printf("%-7d %-23s %-6d| ", i + 1, prod[i].nama, prod[i].stok);
      6. printf("%-7d %-23s %d\n", i + 11, prod[i + 10].nama, prod[i + 10].stok);
      7. }
      8. }
   5. **Fungsi** 
      1. void tambahStok(struct produk prod[], int \*update, int \*barang, int \*stokBaru) {
      2. \*barang = 0;
      3. printf("\nApakah anda ingin menambah stok? (Masukkan 1 jika YA): ");
      4. scanf("%d", update);
      5. if (\*update == 1) {
      6. while (\*barang < 1 || \*barang > MAX) {
      7. printf("Pilih produk: ");
      8. scanf("%d", barang);
      9. if (\*barang < 1 || \*barang > MAX) {
      10. printf("Input tidak valid, silahkan coba lagi.\n");
      11. }
      12. }
      13. printf("Tambahkan stok: ");
      14. scanf("%d", stokBaru);
      15. prod[\*barang - 1].stok += \*stokBaru;
      16. printf("\nStok %s diperbarui: %d", prod[\*barang - 1].nama, prod[\*barang - 1].stok);
      17. }
      18. }
   6. **F**
      1. void rekap(struct produk prod[], int \*totalRekap, int \*rekapDiskon) {
      2. \*totalRekap = 0;
      3. printf("\nREKAP PENJUALAN\n\n");
      4. printf("KODE\tNAMA PRODUK\t\tTERJUAL\t\tHARGA\t\tSUBTOTAL\n");
      5. for (int i = 0; i < MAX; i++) {
      6. printf("%-7d %-23s %-15d Rp. %-11d Rp. %d\n", i + 1, prod[i].nama, prod[i].terjual, prod[i].harga, prod[i].harga \* prod[i].terjual);
      7. \*totalRekap += prod[i].harga \* prod[i].terjual;
      8. }
      9. printf("\nTOTAL\t\t: Rp. %d", \*totalRekap);
      10. printf("\nTOTAL DISKON\t: Rp. %d", \*rekapDiskon);
      11. printf("\nTOTAL BERSIH\t: Rp. %d", \*totalRekap - \*rekapDiskon);
      12. }
   7. **F**
      1. void namaKasir (char kasir[20]) {
      2. printf("\nMasukkan nama kasir: ");
      3. getchar();
      4. gets(kasir);
      5. }
   8. **F**
      1. void tampilkanProduk(struct produk prod[]) {
      2. printf("\nDAFTAR PRODUK\n\n");
      3. printf("KODE\tNAMA PRODUK\t\tHARGA\t\tKODE\tNAMA PRODUK\t\tHARGA\n");
      4. for (int i = 0; i < MAX / 2; i++) {
      5. printf("%-7d %-23s Rp. %-10d| ", i + 1, prod[i].nama, prod[i].harga);
      6. printf("%-7d %-23s Rp. %-12d\n", i + 11, prod[i + 10].nama, prod[i + 10].harga);
      7. }
      8. }
   9. **F**
      1. void pilihProduk(int \*barang, struct produk prod[]) {
      2. \*barang = 0;
      3. while (\*barang < 1 || \*barang > MAX) {
      4. printf("Pilih produk: ");
      5. scanf("%d", barang);
      6. if (\*barang < 1 || \*barang > MAX) {
      7. printf("Input tidak valid, silahkan coba lagi.\n");
      8. }
      9. else if (prod[\*barang - 1].stok == 0) {
      10. printf("Stok produk habis, silahkan pilih produk lain.\n");
      11. \*barang = 0;
      12. }
      13. }
      14. }
   10. **F**
       1. void masukkanJumlah(struct produk prod[], int \*jumlah, int \*barang) {
       2. \*barang -= 1;
       3. \*jumlah = 0;
       4. while (\*jumlah < 1 || \*jumlah > prod[\*barang].stok) {
       5. printf("Masukkan jumlah produk : ");
       6. scanf("%d", jumlah);
       7. if (\*jumlah < 1 || \*jumlah > prod[\*barang].stok) {
       8. printf("Stok: %d\n", prod[\*barang].stok);
       9. printf("Stok tidak mencukupi, silahkan coba lagi.\n");
       10. }
       11. }
       12. prod[\*barang].stok -= \*jumlah;
       13. prod[\*barang].pesan += \*jumlah;
       14. prod[\*barang].terjual += \*jumlah;
       15. }
   11. **F**
       1. void hitungDiskon (struct produk prod[], int \*total, int \*diskon) {
       2. \*diskon = 0;
       3. \*total = 0;
       4. for (int i = 0; i < MAX; i++) {
       5. \*total += prod[i].harga \* prod[i].pesan;
       6. }
       7. if (\*total >= 250000 && \*total < 500000) {
       8. \*diskon = 5;
       9. }
       10. else if (\*total >= 500000) {
       11. \*diskon = 10;
       12. }
       13. }
   12. **F**
       1. void detailPesanan (struct produk prod[], int \*total, int \*diskon) {
       2. printf("\nDETAIL PESANAN\n");
       3. for (int i = 0; i < MAX; i++) {
       4. if(prod[i].pesan > 0)
       5. printf("%2d. %s\n    %-2d x Rp.%-21d Rp. %d\n", i + 1, prod[i].nama, prod[i].pesan, prod[i].harga, prod[i].harga \* prod[i].pesan);
       6. }
       8. printf("---------------------------------------------");
       9. printf("\nSUBTOTAL\t\t\t: Rp. %d", \*total);
       10. printf("\nDISKON %d%%\t\t\t: Rp. %d", \*diskon, (\*total) \* (\*diskon)/100);
       11. printf("\nTOTAL\t\t\t\t: Rp. %d", (\*total) \* (100 - \*diskon)/100);
       12. }
   13. **F**
       1. void beliLagi(int \*tambah) {
       2. printf("\n\n1. Tambah\n");
       3. printf("2. Hapus\n");
       4. printf("3. Selesai\n");
       5. while (\*tambah < 1 || \*tambah > 3) {
       6. printf("Pilih menu: ");
       7. scanf("%d", tambah);
       8. if (\*tambah < 1 || \*tambah > 3) {
       9. printf("Input tidak valid, silahkan coba lagi.\n");
       10. }
       11. }
       12. }
   14. **A**
       1. void hapusProduk(struct produk prod[], int \*hapus, int \*jumlahHapus) {
       2. \*hapus = 0;
       3. \*jumlahHapus = 0;
       4. while (\*hapus < 1 || \*hapus > MAX) {
       5. printf("Pilih produk yang ingin dihapus: ");
       6. scanf("%d", hapus);
       7. if (\*hapus < 1 || \*hapus > MAX || prod[\*hapus - 1].pesan == 0) {
       8. printf("Input tidak valid, silahkan coba lagi.\n");
       9. \*hapus = 0;
       10. }
       11. }
       12. \*hapus -= 1;
       13. while (\*jumlahHapus < 1 || \*jumlahHapus > prod[\*hapus].pesan) {
       14. printf("Masukkan jumlah produk yang ingin dihapus: ");
       15. scanf("%d", jumlahHapus);
       16. if (\*jumlahHapus < 1 || \*jumlahHapus >  prod[\*hapus].pesan) {
       17. printf("Input melebihi pesanan, silahkan coba lagi.\n");
       18. }
       19. }
       20. prod[\*hapus].stok += \*jumlahHapus;
       21. prod[\*hapus].terjual -= \*jumlahHapus;
       22. prod[\*hapus].pesan -= \*jumlahHapus;
       23. }
   15. **A**
       1. void metodePembayaran (int \*pembayaran) {
       2. \*pembayaran = 0;
       3. printf("\nMETODE PEMBAYARAN\n");
       4. printf("1. Tunai\n");
       5. printf("2. QRIS\n");
       6. printf("3. Kredit\n");
       7. while (\*pembayaran < 1 || \*pembayaran > 3) {
       8. printf("Pilih metode pembayaran: ");
       9. scanf("%d", pembayaran);
       10. if (\*pembayaran < 1 || \*pembayaran > 3) {
       11. printf("Input tidak valid, silahkan coba lagi.\n");
       12. }
       13. }
       14. }
   16. **A**
       1. void konfirPembayaran (int \*uang, int \*total, int \*pembayaran, int \*diskon) {
       2. \*uang = 0;
       3. if (\*pembayaran == 1) {
       4. while (\*uang < \*total - \*diskon) {
       5. printf("Masukkan uang: ");
       6. scanf("%d", uang);
       8. if (\*uang < \*total - \*diskon) {
       9. printf("Uang tidak cukup, silahkan coba lagi.\n");
       10. }
       11. }
       12. }
       13. else {
       14. while (\*uang != 1) {
       15. printf("Konfirmasi pembayaran (Masukkan 1 jika SELESAI): ");
       16. scanf("%d", uang);
       18. if (\*uang != 1) {
       19. printf("Pembayaran tidak valid, silahkan coba lagi.\n");
       20. }
       21. }
       22. \*uang = \*total - \*diskon;
       23. }
       24. }
   17. **A**
       1. void pembayaran\_kembalian (int \*pembayaran, int \*uang, int \*total, int \*diskon) {
       2. if (\*pembayaran == 1) {
       3. printf("\nTUNAI\t\t\t\t: Rp. %d", \*uang);
       4. }
       5. else if (\*pembayaran == 2) {
       6. printf("\nQRIS\t\t\t\t: Rp. %d", \*total);
       7. }
       8. else if (\*pembayaran == 3) {
       9. printf("\nKREDIT\t\t\t\t: Rp. %d", \*total);
       10. }
       11. printf("\nKEMBALI\t\t\t\t: Rp. %d\n", \*uang - (\*total - \*diskon));
       12. }
   18. **A**
       1. void tampilkanJam () {
       2. struct tm \*jam;
       3. time\_t waktu;
       4. waktu = time(NULL);
       5. jam = localtime(&waktu);
       6. printf("TANGGAL\t: %02d-%02d-%-14d JAM : %02d:%02d:%02d\n",
       7. jam->tm\_mday, jam->tm\_mon + 1, jam->tm\_year + 1900,
       8. jam->tm\_hour, jam->tm\_min, jam->tm\_sec);
       9. }
   19. **A**
       1. void cetakStruk (struct produk prod[], int \*total, int \*uang, int \*pembayaran, int \*diskon, int \*rekapDiskon, char kasir[20]) {
       2. printf("\n%28s", "DUROMART\n");
       3. printf("%36s", "Jl. Sidotopo Wetan No. 15\n");
       4. printf("%33s", "Telp: 085730565400\n");
       5. printf("---------------------------------------------\n");
       6. tampilkanJam();
       7. printf("KASIR\t: %s\n", kasir);
       8. printf("---------------------------------------------");
       9. detailPesanan (prod, total, diskon);
       10. pembayaran\_kembalian(pembayaran, uang, total, diskon);
       11. printf("---------------------------------------------\n");
       12. printf("  TERIMA KASIH TELAH BERBELANJA DI DUROMART  \n");
       13. for (int i = 0; i < MAX; i++) {
       14. prod[i].pesan = 0;
       15. }
       16. \*rekapDiskon += (\*total) \* (\*diskon)/100;
       17. }
   20. **A**
3. int main() {
4. int program = 1;
5. char kasir[20];
6. int barang;
7. int jumlah;
8. int total;
9. int diskon;
10. int hapus;
11. int jumlahHapus;
12. int pembayaran;
13. int uang;
14. int update;
15. int stokBaru;
16. int totalRekap;
17. int rekapDiskon = 0;
19. struct produk prod[MAX];
20. listProduk(prod);
22. while (program == 1) {
23. int tambah = 1;
24. int menu = 0;
25. tampilkanMenu(&menu);
26. if (menu == 1) {
27. namaKasir (kasir);
28. tampilkanProduk(prod);
29. pilihProduk(&barang, prod);
30. masukkanJumlah(prod, &jumlah, &barang);
31. hitungDiskon (prod, &total, &diskon);
32. detailPesanan(prod, &total, &diskon);
33. while (tambah == 1) {
34. tambah = 0;
35. beliLagi(&tambah);
36. if (tambah == 1) {
37. tampilkanProduk(prod);
38. pilihProduk(&barang, prod);
39. masukkanJumlah(prod, &jumlah, &barang);
40. hitungDiskon (prod, &total, &diskon);
41. detailPesanan(prod, &total, &diskon);
42. }
43. else if (tambah == 2) {
44. hapusProduk(prod, &hapus, &jumlahHapus);
45. hitungDiskon (prod, &total, &diskon);
46. detailPesanan(prod, &total, &diskon);
47. tambah = 1;
48. }
49. else if (tambah == 3) {
50. metodePembayaran(&pembayaran);
51. konfirPembayaran(&uang, &total, &pembayaran, &diskon);
52. cetakStruk (prod, &total, &uang, &pembayaran, &diskon, &rekapDiskon, kasir);
53. }
54. }
55. }
56. else if (menu == 2) {
57. cekStok(prod);
58. tambahStok(prod, &update, &barang, &stokBaru);
59. }
60. else if (menu == 3) {
61. rekap(prod, &totalRekap, &rekapDiskon);
62. }
63. printf("\nKembali ke menu utama? (Masukkan 1 jika YA): ");
64. scanf("%d", &program);
65. }
66. printf("\nTerima kasih telah menggunakan Duromart!\n");
67. return 0;
68. }