



Program Pembagian Kue / Makanan Untuk Arisan

Dosen Pengampu: Sahala Wahyu
Wardana, S.Si., M.Cs.

By_Kelompok 7



Anggota kelompok

Febrian Zaki Hidayatullah	(L0325045)
Duta Lathiif Haqim	(L0325021)
Naila Bunga Lestari	(L0325008)
Kayla Azzura Salim	(L0325039)

Latar Belakang

Alasan program ini dibuat yaitu :

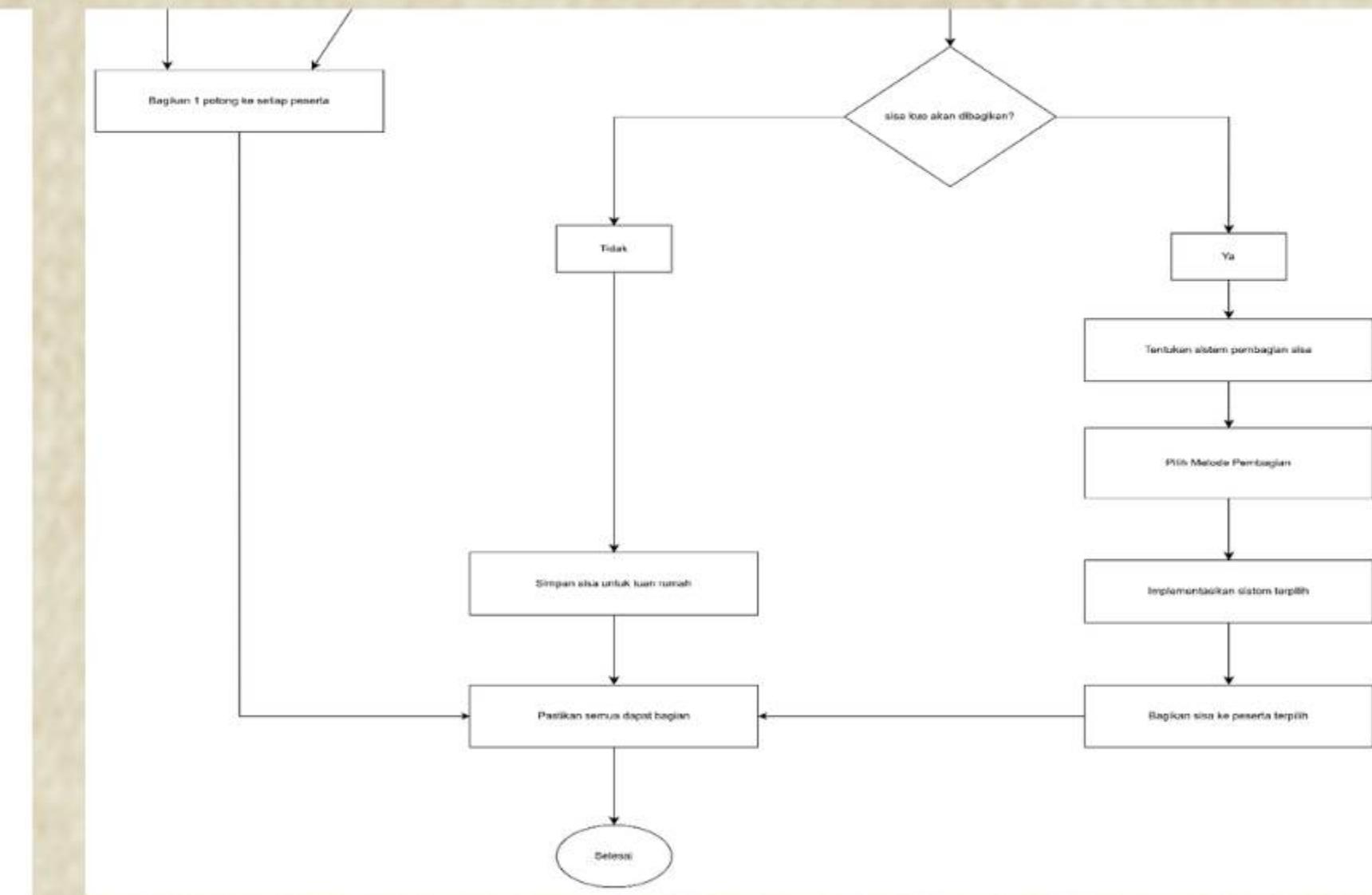
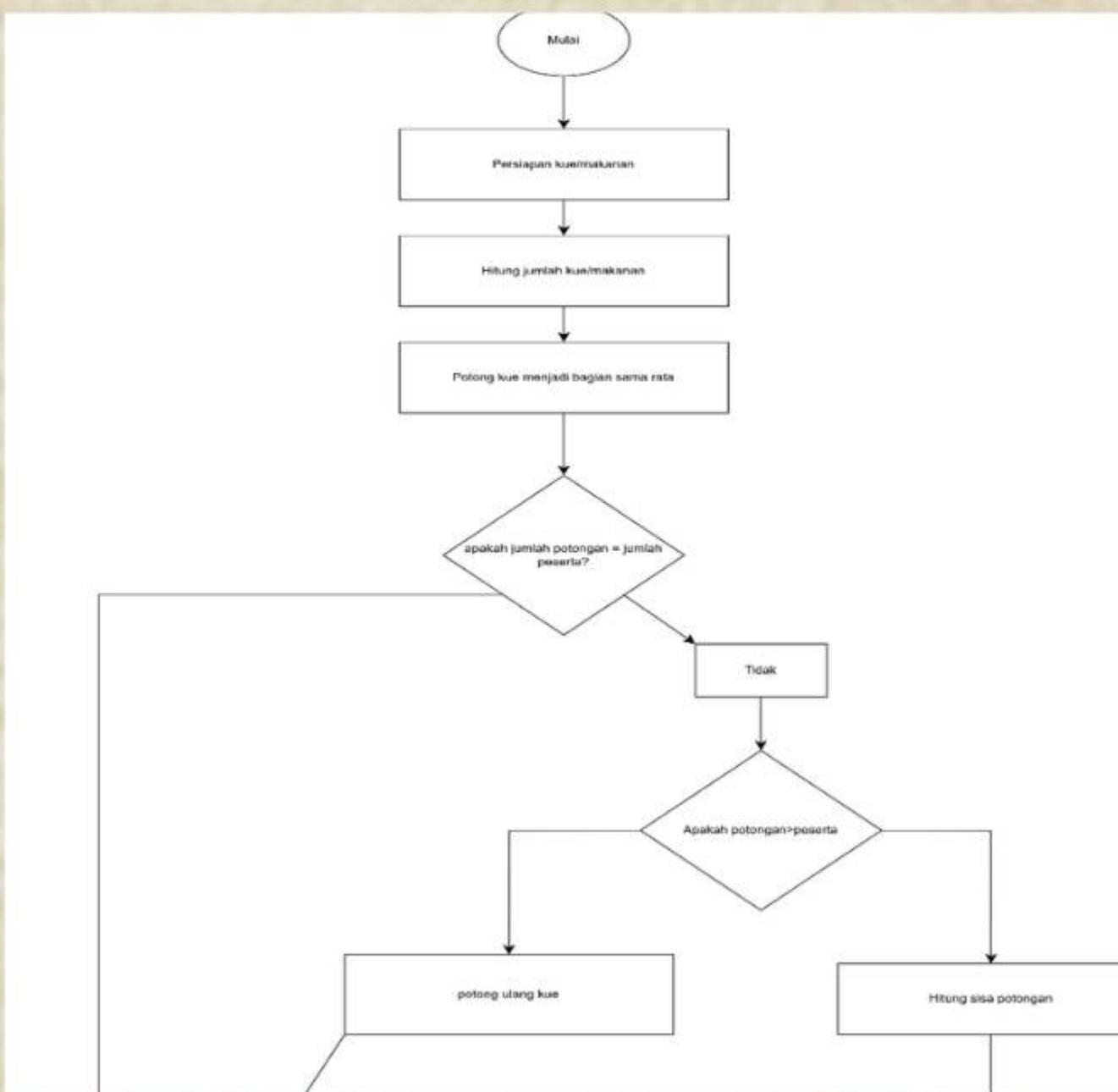
1. Masalah Nyata: Dalam arisan, pembagian makanan sering tidak merata dan menimbulkan ketidakpuasan.
2. Penyebab Teknis: Ketidakseimbangan antara jumlah potongan kue dan jumlah undangan, serta adanya peserta yang terlambat.

Dampak Sosial: Kurangnya transparansi dalam pembagian dapat mengganggu keharmonisan sosial antar peserta.

Solusi

1. Perhitungan Objektif: Menggunakan logika matematika untuk memastikan setiap orang mendapat porsi yang sama.
2. Manajemen Sisa: Mengatur penanganan sisa makanan dengan mekanisme yang jelas (dibagi rata, diundi, atau diberikan manual).
3. Transparansi: Mencegah kecurigaan karena proses pembagian dilakukan secara terbuka oleh sistem komputer.

Diagram Flowchart



Alur Logika (Flowchart)

1. input Data : Masukkan jumlah peserta dan jumlah potongan kue.
2. Pengecekan : Program membandingkan jumlah potongan dengan peserta.
3. Kondisi Percabangan:
 - Tidak tersisa: Langsung bagikan 1 per orang.
 - Sisa: Bagikan 1 per orang, lalu sisa kue diproses ulang.
 - Kurang : Beri peringatan, lalu opsi tambah potongan atau potong ulang.

Implementasi Teknis

1. mulai

```
int main() {  
    srand((unsigned) time(NULL));
```

2. Persiapkan Kue

```
intf("Masukkan jumlah potongan kue yang dipotong: "  
(scanf("%d", &potongan) != 1) { while (getchar() !=
```

user memasukan jumlah kue yang akan di potong

3. Hitung Jumlah Peserta

```
printf("Masukkan jumlah peserta: ");  
if (scanf("%d", &peserta) != 1) { while (getchar() != '\n'); peserta = 0; }
```

user mamasukan jumlah peserta

4. Potong Kue menjadi Bagian Sama Rata

5. Apakah Jumlah Potongan = Jumlah Peserta?

```
if (potongan == peserta) {
```

Program memeriksa apakah potongan tepat sama dengan peserta.

Ya → lanjut ke langkah 6 .

Tidak → lanjut ke langkah 7

6. Bagikan 1 Potong ke Setiap Peserta (Jika potongan = peserta)

distribute_one_each(dist, peserta);

Semua peserta langsung menerima 1 potong

7. Apakah Potongan > Peserta?

} else if (potongan > peserta) {

Ya → lanjut ke langkah 8

Tidak → lanjut ke langkah 15

8. Hitung Sisa Potongan

int sisa = potongan - peserta;

Menghitung potongan yang tersisa setelah pembagian awal

9. Bagikan 1 Potong ke Setiap Peserta

distribute_one_each(dist, peserta);

10. Kemana Sisa Kue Akan Dibagikan?

```
printf("\nPilih metode pembagian sisa:\n");
printf("1 - Dibagikan merata (satu per peserta berulang)\n");
printf("2 - Diundi secara acak\n");
printf("3 - Diberikan ke peserta tertentu (input manual)\n");
```

memilih sistem pembagian lanjut ke langkah 11

langsung lanjut ke langkah 18 (Pastikan semua dapat bagian).

11. Tentukan Sistem Pembagian Sisa

```
int main() {
    srand((unsigned) time(NULL));
```

User memilih metode pembagian sisa: merata, undian, atau spesifik

12. Pilih Metode Pembagian

```
printf("Pilih (1/2/3): ");
if (scanf("%d", &metode) !=
```

Program membaca pilihan metode user.

13. Implementasikan Sistem Terpilih

```
if (metode == 1) {
    distribute_extra_merata(dist, peserta, sisa);
} else if (metode == 2) {
    distribute_extra_undian(dist, peserta, sisa);
} else {
    distribute_extra_specific(dist, peserta, sisa);
}
print_distribution(dist, peserta);
```

Program menjalankan metode sesuai pilihan user.

14. Bagikan Sisa ke Peserta Terpilih

```
if (metode == 1) {
    distribute_extra_merata(dist, peserta, sisa);
} else if (metode == 2) {
    distribute_extra_undian(dist, peserta, sisa);
} else {
    distribute_extra_specific(dist, peserta, sisa);
}
print_distribution(dist, peserta);
```

15. Jika Potongan < Peserta → Potong Utang Kue / Tambah Potongan

```
else if (potongan > peserta) {  
    // Ada sisa potongan  
    int sisa = potongan - peserta;  
    distribute_one_each(dist, peserta);  
    printf("Jumlah potongan lebih dari peserta. Setiap peserta mendapat 1. Sisa: %d\n", sisa);
```

User menambah potongan → lanjut ke langkah 16.

User tidak menambah → lanjut ke langkah 17

16. Tambah Potongan Bila Dibutuhkan

potongan += tambah;

17. Bagikan Secara Acak Jika Tetap Kurang

distribute extra undian(dist, peserta, potongan);

Jika potongan tetap tidak cukup, pembagian dilakukan secara acak.

18. Pastikan Semua Dapat Bagian

print_distribution(dist, peserta);

Hasil pembagian akhir ditampilkan untuk memastikan semuanya menerima bagian.

19. Selesai

```
free(dist);  
printf("Selesai. Terima kasih.\n");  
return 0;
```

program selesai

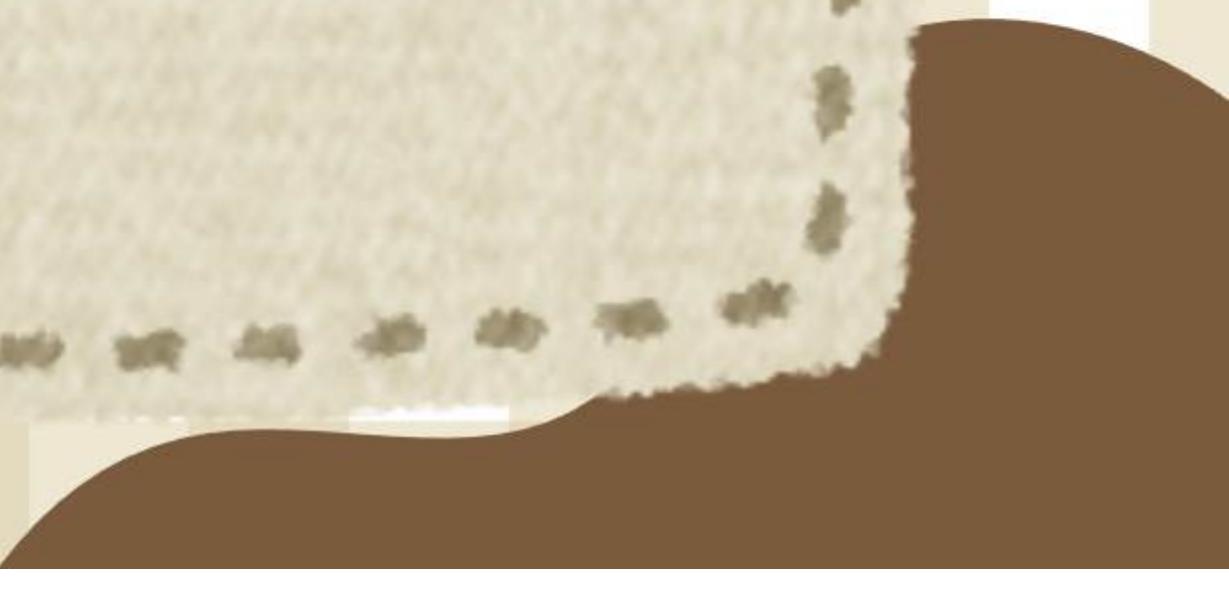
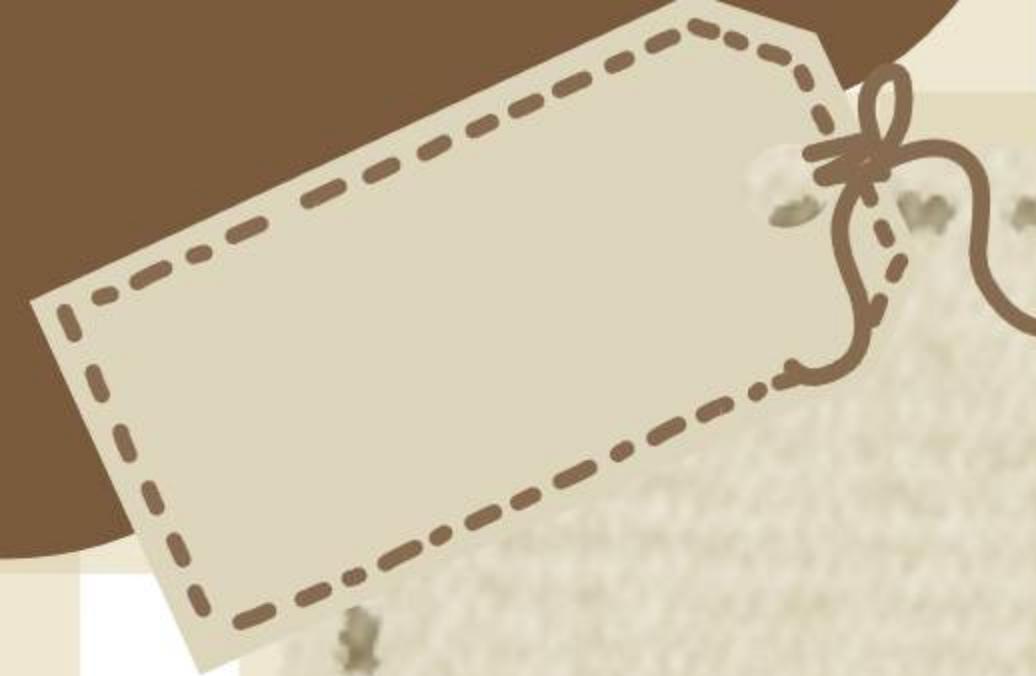
Output

```
//Output
/*== Aplikasi Pembagian Kue ===
Masukkan jumlah peserta: 12
Masukkan jumlah potongan kue yang dipotong: 12
Jumlah potongan sama dengan jumlah peserta -> langsung dibagikan 1 potong tiap peserta.

--- Hasil Pembagian ---
Peserta 1 : 1 potong
Peserta 2 : 1 potong
Peserta 3 : 1 potong
Peserta 4 : 1 potong
Peserta 5 : 1 potong
Peserta 6 : 1 potong
Peserta 7 : 1 potong
Peserta 8 : 1 potong
Peserta 9 : 1 potong
Peserta 10 : 1 potong
Peserta 11 : 1 potong
Peserta 12 : 1 potong
-----
Selesai. Terima kasih.*/
```

Kesimpulan

- Program berhasil memberikan solusi objektif atas masalah ketidakmerataan pembagian makanan.
- Penerapan Ilmu : Berhasil mengimplementasikan variabel, logika if-else, looping, dan array dalam kasus nyata.
- Nilai Tambah: Melatih pola pikir analitis untuk menyelesaikan masalah sosial sehari-hari



**Thank
You**